Online Machine Learning Framework for Budgeted Bandits with an Option of Giving Up

by

Sharoff Pon Kumar

A THESIS Submitted in Partial Fulfillment of the
Requirements for the Degree of

MASTER OF SCIENCE

in the Department of Computer Science

Online Machine Learning Framework for Budgeted Bandits with an Option of Giving Up

by

Sharoff Pon Kumar

Supervisory Committee

Dr. Nishant Mehta, Supervisor
(Department of Computer Science)

Dr. Jianping Pan, Departmental Member
(Department of Computer Science)

Supervisory Committee

Dr. Nishant Mehta, Supervisor
(Department of Computer Science)

Dr. Jianping Pan, Departmental Member
(Department of Computer Science)

ABSTRACT

We study an online learning problem where the game proceeds in epochs and an agent takes an action in each epoch. Depending upon the action, the agent receives a stochastic reward and the time taken for completing an epoch also depends on a stochastic delay. The agent can only take a new action once the previous action is completed. The game ends once the total allotted time budget runs out. The goal of the agent is to maximize its cumulative reward over a fixed budget. However, the agent can also "give up" on a action to optimize the time budget, which prevents the agent from collecting that reward associated with that action; it can then choose another action. We model this problem as a variant of multi-armed bandits problem having stochastic reward and stochastic resource consumption with a fixed global budget. For this problem, we first establish that the optimal arm is the arm that maximizes the ratio of the expected reward of the arm to the expected waiting time before the agent sees the reward due to pulling that arm. We then propose an upper confidence bound-based algorithm Wait-UCB using a novel upper confidence bound developed on this ratio which attains a logarithmic, problem-dependent regret bound with an improved dependence on the problem dependent parameters compared to previous works. We conduct simulations on the proposed algorithm in various problem configuration comparing Wait-UCB against state-of-the-art algorithms, verifying the effectiveness of our proposed algorithm. We then study this problem with additional feedback, more than mere bandit feedback, where the agent observes the reward of the actions having shorter waiting time; we call this type of feedback "leftward chain feedback". For this problem with additional feedback, we develop a novel upper confidence bound-based algorithm, Wait-2 Learn UCB, which guarantees logarithmic, problem-dependent regret bound. However, our regret bound does not yet show any improvement over the regret bound for Wait-UCB.

Table of Contents

List of Figures

Acknowledgements

First of all, I would like to thank Dr. Nishant Mehta for the supportive and encouraging guidance in research throughout my degree. I am thankful to have learnt some of the best practices which has made me the researcher I am today. With his infectious passion for research, he has always inspired me to think about some of the ambitious problems above and beyond. As an advisor, he has helped me with thought provoking feedback and helped me find the right research question to unblock myself. These learning has not only helped me during this research project but will inspire me in the future as well.

I would like to thank my mother Geetha who has always made sure that I get good education despite the challenges and circumstances. I want to thank a special person my brother Parosh (as I call him) for always being there for me since my childhood, for all the personal support during my ups and downs in my life and also for being strong support for our family. His guidance, suggestions and thoughtful discussions have always helped me throughout my life. I want to thank my father for being there for our family. Thanks to Mohana Krishnan, Kanthimathi, Ebenezer and Annie for their support.

Also, I would like to thank Dr. Jianping Pan for being a part of the supervisory committee. I am thankful to all my friends, colleagues in our lab for all the useful discussion. Always, I am grateful to Lord Almighty for helping me throughout my degree, for all the opportunities and success in my life.

Dedication

To my Mother, Brother & God.

Preface

The thesis consist of two parts. The first part of the thesis, which is discussed in Chapter 2 to 5 covers the motivation, problem setting and a new algorithm for solving this problem along with empirical evaluation and theoretical analysis. The contributions of this part have been published in:

Referred Conference Publication

Sharoff P, Nishant Mehta, and Ravi Ganti, A Farewell to Arms: Sequential Reward Maximization on a Budget with a Giving Up Option, *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2020 (Sharoff et al., 2020)

The second part of the work, explained in Chapter 6, covers a new algorithm to solve the problem introduced in previous chapters for the case of receiving additional feedback.

Chapter 1

Introduction

Multi-armed bandits problems are sequential decision making problems significantly studied for taking decisions under uncertainty over time in order to maximize the total collected reward. These problems pose a challenge of handling the dilemma between exploration and exploitation. It is not always the case that taking the decision with high payout provides the highest total collected reward in the end. In practical situations, the learner has to handle multiple constraints according to the task it intends to solve. Sometimes, the agent cannot observe the reward as soon as it takes an action and in some cases there is consumption of resources that is associated with a decision. We solve a variant of multi-armed bandits problem where the agent has to handle both a resource consumption associated in taking a decision and the delay in getting a reward. In each epoch, the agent takes an action and waits for the reward to be returned which depends on a delay distribution. While the agent is waiting, it cannot take any other action and it has only a fixed time budget. The goal of the agent is to maximize the total reward that could be accumulated within this fixed time budget.

Consider, for example, a model of crowdsourcing where an employer submits tasks to a crowdsourcing platform which has a large pool of workers. The employer wishes

to have as many tasks as possible completed under a fixed time budget. However, for a given type of task, the efficiency of workers can vary significantly. While some omniscient actor could dispatch tasks only to fast workers, a typical employer would not know the efficiency of different workers on the task at hand. If a randomly selected worker is fast, it can be worthwhile to wait for the worker to complete a job; however, if a worker takes too long to complete a task, it can be advantageous for the employer to terminate the assignment and reassign the task to a new worker. How long should the employer wait for an assigned task to be completed before reassigning the task to a new worker? This problem fits into a new multi-armed bandit framework, *Farewell to Arms* (F2A), that introduces the idea of "giving up" on an action (arm) that takes too long to return a reward; for a given arm, we view choosing the waiting time itself as pulling a kind of "micro-arm". As we explain in Section 3.2, other applications include hyperparameter tuning, repeated second-price auctions with participation costs, and computational advertising.

Informally, the F2A framework is a variant of the classical stochastic K-armed bandit problem framework, but with an added twist of a stochastic resource consumption. For simplicity, consider first the case of the F2A framework with only a single arm. Upon pulling the single arm the learning agent gets some stochastic reward, not instantaneously, but only after a stochastic delay[1], say τ. There could be cases where the delay is extremely large. To incorporate this notion, the learning agent is not only required to choose an arm, but is also required to commit to a waiting time j. The learning agent receives a reward if $j \geq \tau$, i.e., the agent is willing to wait until the reward arrives, and gets a reward of 0 otherwise. Hence, the F2A framework can be thought of as a 2-level stochastic multi-armed bandit framework with K "macro-arms" and (say) D "micro-arms" for each of the K macro-arms. The

[1]The stochastic rewards and delays can be dependent.

micro-arms capture the willingness of the learning agent to wait and hence bound the total amount of time that can be consumed by a pull, while the macro-arms capture the goodness of a certain arm.

An F2A problem can be cast into the more general bandits with knapsack (BwK) framework (Badanidiyuru et al., 2013) by considering the latter with only a single resource. However, whereas Badanidiyuru et al. (2013) established worst-case regret bounds for the BwK problem, in this work we establish (logarithmic) problem-dependent regret bounds. Such logarithmic problem-dependent regret bounds tend to be sharper and better capture problem complexity in the standard stochastic multi-armed bandit case when the gap between the best and the second best arm is large. The pioneering work of Flajolet and Jaillet (2017) established, for their algorithm UCB-Simplex, logarithmic problem-dependent regret bounds (growing logarithmically in the number of rounds) that also apply to F2A problems. However, the new algorithm we develop — Wait-UCB — is based on a rather different upper confidence bound than the one used by UCB-Simplex; moreover, the regret bounds we prove for Wait-UCB are often better than those of Flajolet and Jaillet (2017) and also those derived by Xia et al. (2016) for their algorithm Budget-UCB. Furthermore, we also show that Wait-UCB has better empirical performance than UCB-Simplex and Budget-UCB (Xia et al., 2016). Many important problems (as discussed in Section 3.2) fall in the more specialized F2A framework and hence this framework, despite being a special case of the BwK framework, demands a specialized treatment and better algorithms.

Our core contributions are as follows:

- We introduce the Farewell to Arms framework.
- We show in Section 4.1 that, due to the stochastic consumption of resources in an F2A problem, the right quality measure for an arm is the ratio of expected

reward to expected waiting time.

- We derive in Section 4.2 a novel upper confidence bound for the ratio of mean reward to mean waiting time; using this bound, we design Wait-UCB, a new upper confidence-style algorithm.

- We establish (Section 4.3) a logarithmic problem-dependent regret guarantee for Wait-UCB which is never worse than $\mathcal{O}((D^3/\Delta)\log T)$, where Δ is the gap between the aforementioned ratio for the best arm to the best suboptimal arm. In important regimes for F2A problems (like when mean waiting times for most arms are small), our bound can be D times smaller than the regret bounds for UCB-Simplex (Flajolet and Jaillet, 2017) and Budget-UCB (Xia et al., 2016).

- We provide (Section 5) a detailed experimental study of Wait-UCB, including comparisons to UCB-Simplex and Budget-UCB which suggest that Wait-UCB fares better for F2A problems.

- We derive a novel upper confidence bound in the form of cumulative density function (CDF) to account for the additional feedback observed in the farewell to arms problem and using this bound we developed a new upper confidence style algorithm, Wait-2 Learn UCB in Section 6.2.

- Also, we show a similar logarithmic problem-dependent regret bound guarantee for Wait-2 Learn UCB in Section 6.4 which suffers a regret in the order of $\mathcal{O}\left(\sum_{j|\Delta>0}\frac{D^3}{\Delta}\log T\right)$, where Δ is the gap between the ratio for the best arm to the best sub-optimal arm. Though, this regret bound is not tighter than that of Wait-UCB's, it does manage to match it under the best case scenario, which happens when the last arm D happens to be pulled repeatedly to attain the sufficient sampling. Also, the main contribution of Wait-2 Learn UCB is

its ability to handle the extra feedback without the use of a directed feedback graph.

Chapter 2

Background

On the surface, F2A problems bear close similarity to the problem of learning the optimal waiting time. In the latter problem, studied in detail by Lattimore et al. (2014), in each of a fixed number of rounds a learning agent selects a waiting time. If a stochastic delay exceeds this waiting time, the agent suffers a loss equal to the waiting time plus a fixed cost; otherwise, the agent suffers the stochastic delay itself plus a different fixed cost. While learning an optimal waiting time is a common thread between the work of Lattimore et al. (2014) and our work, there are three key differences: in an F2A problem, *(i)* the time spent affects a budget; *(ii)* there is separate collection of stochastic reward (which is not present at all in the work of Lattimore et al. (2014)); and *(iii)* the game has a random stopping time that depends on all the actions taken, making exploration more challenging. Consequently, the optimal waiting time differs considerably in our setting, instead depending on a ratio of means.

The F2A framework is however very related to the bandits with knapsacks (BwK) setting (Badanidiyuru et al., 2013) (see also the earlier work of Tran-Thanh et al. (2012) and Ding et al. (2013)). A BwK problem is a generalization of the classical multi-armed bandit problem (Lai and Robbins, 1985) in which the learning agent

has finite quantities of a number of resources, and each pull of an arm stochastically consumes each resource while also yielding some stochastic reward (the reward and resource consumptions in each round can be dependent). The game ends once any resource is exhausted. The classical multi-armed bandit problem is recovered by taking time as the single resource, which deterministically decreases by 1 when any arm is pulled (the game ends when the finite time budget is exhausted). F2A problems also can be cast in the BwK setting, now by taking time as a single resource which is consumed stochastically. However, the full BwK setting is so general that the algorithms developed for this setting, and the type of regret guarantees given, differ substantially from the type of guarantees we seek here. In particular, we seek problem-dependent bounds with regret growing logarithmically with the size of the budget. Such bounds previously were obtained by Ding et al. (2013), Xia et al. (2016) and Flajolet and Jaillet (2017), but, as we explain in Section 4.3, in the case of F2A problems our results can be better in important regimes.

Another line of work in the literature that bears a resemblance to the characteristics of the F2A problem are Threshold Bandits (Abernethy et al., 2016) and Firing Bandits (Jain and Jamieson, 2018). In Threshold Bandits by Abernethy et al. (2016), the learning agent only gets a reward if a random value depending on the action is above a certain threshold value. The F2A problem differs from threshold bandits in many ways. In the F2A problem, the desired action taken is tightly linked with both the reward and the cost as the problem deals with a budget constraint which is not the case in threshold bandits. Also, the potential reward is drawn from a distribution and it is not necessarily binary and it can also be dependent on the delay. Moreover, a cyclic permutation assumption is made in the threshold bandits on drawing the threshold values from a set but in the F2A problem, the delay is drawn from a distribution τ. Also, the F2A problem offers additional side information which is discussed

in the later part of the text. These are some of the differences with Threshold Bandits. There is another type of thresholding discussed in the work of Firing Bandits by Jain and Jamieson (2018). In this problem, the learning agent selects a coin and flips it in each round. The learning agent gets a reward if the coin reaches a certain number of successes; otherwise, it gets no reward. We can see that the thresholding aspect of the problem itself varies from the F2A problem. Also, the resource consumption is not directly linked with the action taken in each epoch. Since the problem is designed to maximize the projects getting funded in a crowdfunding platform, the arms representing the projects grows over time. Firing Bandits features a growing set of arms as opposed to fixed set of arms which in a way is closely related to infinite armed bandit problem.

Finally, we mention in passing that there also are weak connections to two other settings. In bandits with delayed feedback (Joulani et al., 2013), feedback from arms can be delayed, but the learning agent can still pull an arm in every round; this difference is critical. In bandits with lock-up periods (Komiyama et al., 2013), feedback is not delayed and an arm can be pulled in each round, but the learning agent experiences "lock-up" periods during which it must constantly pull the same arm. This is similar to our waiting period, but an important difference is that at the end of a waiting period in our setting, the learning agent only receives one reward (possibly equal to zero), whereas in the lock-up period setting, a reward is received in each round.

Chapter 3

Learning Problem

We now formally introduce the Farewell to Arms framework and then show how it captures several important applications.

3.1 A Farewell to Arms game

In the F2A framework, there is a hierarchical set of arms with K macro-arms at the top level and, for each macro-arm, D micro-arms at the next level. We will index macro-arms with $k \in [K] := \{1, 2, \ldots, K\}$ and micro-arms with $j \in [D]$. Associated with each macro-arm k, there is a joint distribution Q_k over $[0, 1] \times [D]$, for a space of rewards $[0, 1]$[1] and a space of delays $[D]$. A game in the F2A framework lasts for T time units and proceeds in epochs. In each epoch $s = 1, 2, \ldots,$

1. The learning agent plays a macro/micro-arm pair $i_s := (k, j) \in [K] \times [D]$.

2. Independently of the learning agent's choice, the stochastic environment draws a potential reward of $V_{k,s} \in [0, 1]$ and a delay of $\tau_{k,s} \in [D]$ from distribution Q_k.[2]

[1] We assume rewards are bounded, in which case it is without loss of generality that we can and will assume that they fall in the unit interval.

[2] We consider a finite number of delay values. The same ideas with minor technical changes can be

3. The agent collects rewards $r_{i_s}^{(s)} := V_{k,s} \cdot \mathbf{1}\left[\tau_{k,s} \leq j\right]$ and consumes $c_{i_s}^{(s)} := \min\{\tau_{k,s}, j\}$ units of time.

Epochs can be of variable length in the F2A framework. This is in contrast to the standard multi-armed bandit framework, where each epoch has unit length. Hence, given a total time budget of T units in the F2A framework there can be a variable number of epochs, whereas in the multi-armed bandit framework there are exactly T epochs.

The goal of the learning algorithm is to maximize reward within the fixed time budget of T. We will study the *pseudo-regret* of the learning algorithm against the best constant policy: the pseudo-regret of a learning algorithm that plays the macro/micro-arm pair sequence $i_1, i_2, \ldots$ is

$$R_T := \max_{(k,j)\in[K]\times[D]} \mathsf{E}\left[\sum_{s=1}^{L_{k,j}} r_{k,j}^{(s)}\right] - \mathsf{E}\left[\sum_{s=1}^{L} r_{i_s}^{(s)}\right]; \tag{3.1}$$

where $L_{k,j}$ and L are the random stopping times when playing the macro/micro-arm pair sequence $((k,j),(k,j),\ldots,))$ and $(i_1, i_2, \ldots)$ respectively. Intuitively, it seems that an optimal (constant) policy is one that maximizes the average reward obtained per unit time. In Theorem 1 we show that this is indeed true and that the *ratio estimator* $\frac{\mathsf{E}[r]}{\mathsf{E}[c]}$, where r is the reward and c is the amount of resource consumed when a certain macro/micro-arm pair is pulled, is indeed the right estimator to optimize for.

applied for the case when the number of delay values is finite but the set of delays is not necessarily equal to $[D]$.

3.2 Applications of the F2A framework

In addition to the crowdsourcing example mentioned earlier, many other applications fit into the F2A framework. We now present a few of them.

Repeated second price auctions. In a repeated, sealed, second price auction (Weed et al., 2016) with participation costs (Gal et al., 2007; Stegeman, 1996; McAfee and McMillan, 1987; Samuelson, 1985), the goal is to maximize the expected cumulative reward given a budget of B dollars. In each round s of the auction, a bidder pays a flat (positive) participation cost of c dollars and selects a bid $i_s \in \{b_1, b_2, \ldots, b_D\}$ along with the other competing bidders; the bidder wins the auction if their bid was the highest. If the bidders wins the auction, they get a reward of V_s and consume a budget of $c + M_s$ dollars, where M_s is the second highest bid. If the bidder loses the auction, their reward is 0 but they consume c dollars of their budget. The game ends once the budget is no longer positive. Under appropriate stochastic assumptions on the items and bids — namely, that the items are drawn i.i.d. (so that the utilities V_s are i.i.d.) and that the highest bids M_s among the other bidders also are i.i.d.[3] — this problem can be cast into the F2A framework where there is a single macro-arm and the D micro-arms are the values of the bids.

k-fold cross-validation. Consider the problem of performing hyperparameter selection via k-fold cross-validation (CV). In k-fold CV, we are required to run a learning algorithm to convergence a very large number of times. For each of a typically large number of hyperparameter configurations, we need to run the learning algorithm on each of k subsets of the training data. Each execution can take a different amount of time to converge, and with random initialization the runtime and also the model

[3] We allow V_s and M_s to be dependent.

learned by the algorithm are stochastic. Suppose that we have a fixed time budget and view the quality of the model learned as potential reward (high quality solution meaning a large reward); then a natural goal is to find a set of hyper-parameters that are near-optimal. A natural formulation of this problem is to cast it as a pure-exploration multi-armed bandit problem (Li et al., 2017). In this paper, we cast the k-fold CV problem as a regret minimization problem, inspired by a similar regret minimization approaches used in bandit convex optimization (Agarwal et al., 2011). In practice, k-fold CV is done with an implicit time budget constraint, where long-running experiments are terminated, receiving a reward of 0, and the experiments are re-started from a different parameter configuration. This practical consideration means that we want to maximize the total cumulative reward under a given total time budget and hence makes this problem fit well into the F2A framework: the macro-arms are the various parameter configurations, and the micro-arms are the amount of time/computational resources we are willing to allocate for different experiments.

Computational advertising. In computational advertising, a publisher wants to show an ad from an inventory of ads. When a user sees a published ad, the user might be interested in it but might not click on the ad immediately. In such cases, there is an economic incentive for the publisher to display the ad for multiple time periods and wait for a response from the user rather than switch to a different ad immediately. However, at the same time the publisher (learning agent) would like to minimize their regret of not showing the best ad. This problem can be cast in the F2A framework, where the macro-arms are the various ads and the micro-arms are the different durations for which the publisher is willing to wait.

Chapter 4

Algorithm and Guarantees

4.1 A ratio estimator for F2A problems

In this section, we find a suitable metric that captures both the reward and resource aspects of F2A problems. An upper confidence bound for this metric will be key in designing the Wait-UCB algorithm in Section 4.2. For notational convenience, in this section we only consider the case of one macro-arm, so $K = k = 1$ (but of course with multiple micro-arms).

Given a finite time budget and full knowledge of the data-generating distribution, which constant policy maximizes expected cumulative reward? As pulls can have stochastic extent, intuitively this policy should always pull the micro-arm that maximizes the ratio of expected reward to expected waiting time. Our first main result gives formal backing to this intuition.

Theorem 1 *Let $(V_1, \tau_1), (V_2, \tau_2), \ldots$ be i.i.d. according to a joint distribution Q over $[0, 1] \times [D]$. Consider the constant policy which pulls micro-arm j in each epoch, so that in a given epoch s this arm consumes (i.e., waits for) $c_j^{(s)} := \min\{j, \tau_s\}$ units of time and collects reward $r_j^{(s)} := \mathbf{1}[j \geq \tau_s] \cdot V_s$.*

Then the total cumulative reward collected by this constant policy under a time

budget of T is equal to

$$\mathsf{E}\left[\sum_{s=1}^{L} r_j^{(s)}\right] = T \cdot \frac{\mathsf{E}[r_j^{(1)}]}{\mathsf{E}[c_j^{(1)}]} + A_j$$

for some constant $A_j \in [0, j]$, where $L = \max\{n : \sum_{s=1}^{n} c_j^{(s)} \leq T\}$ is the last epoch before the game ends.

Proof of Theorem 1

The proof of Theorem 1 relies on the following two fundamental results.

Theorem 2 (Wald's identity (Blackwell, 1946)) *Let $X_1, X_2, \ldots, X_n$ be i.i.d. random variables having finite mean (so $\mathsf{E}\big[|X_1|\big] < \infty$), and L be a stopping time with respect to the filtration $\mathcal{F}_n$ (i.e., $\{L \leq n\} \in \mathcal{F}_n \quad \forall n \in \mathbb{N}$) satisfying $\mathsf{E}[L] < \infty$. Then*

$$\mathsf{E}[X_1 + \ldots + X_L] = \mathsf{E}[L] \cdot \mathsf{E}[X_1].$$

Theorem 3 (Doob's optional stopping theorem (Grimmett et al., 2001)) *Let M_n be a martingale with respect to the filtration $\mathcal{F}_n$ and let L be a stopping time. Suppose the following three conditions hold:*

(a) $P(L < \infty) = 1$;

(b) $\mathsf{E}\big[|M_L|\big] < \infty$;

(c) $\mathsf{E}\big[M_n \cdot \mathbf{1}\,[L > n]\big] \to 0$ *as* $n \to \infty$.

Then $\mathsf{E}[M_L] = \mathsf{E}[M_0]$.

With the above results at hand, we now prove Theorem 1.

PROOF (OF THEOREM 1) Recall that, for any epoch s, we have $c_j^{(s)} = \min\{j, \tau_s\} \leq j$. Let w be the mean waiting time for micro-arm j, so that

$$w := \mathsf{E}[c_j^{(1)}] = \mathsf{E}\Big[\min\{j, \tau_1\}\Big] = \sum_{k=1}^{j-1} k \cdot \Pr(\tau_1 = k) + j \cdot \Pr(\tau_1 \geq j).$$

Let $S_n := \sum_{s=1}^{n} c_j^{(s)}$ be the sum of the waiting times when arm j is pulled for the first n epochs, and let $S_0 := 0$.

Because the waiting times $c_j^{(s)}$ are stochastic, the number of epochs before the game ends (i.e. before the budget is depleted) is random. Let L be the stopping time with respect to the filtration $(\mathcal{F}_n)_{n\geq 0}$, defined as

$$L = \max\{n : S_n \leq T\}. \tag{4.1}$$

Now, from Wald's identity (Theorem 2), the total cumulative reward obtained by the constant policy that always pulls arm j under a time budget T is equal to

$$\mathsf{E}\left[\sum_{s=1}^{L} r_j^{(s)}\right] = \mathsf{E}[r_j^{(1)}] \cdot \mathsf{E}[L], \tag{4.2}$$

where we recall that $r_j^{(s)} = V_s \cdot \mathbf{1}\left[\tau_s \leq j\right]$ is the reward for pulling arm j in epoch s.

Define a martingale $(M_n)_{n\geq 0}$ by $M_n := S_n - n \cdot w$. Since each $c_j^{(s)}$ lies in the interval $[1, j]$, the conditions of Doob's optional stopping theorem (Theorem 3) hold, and so $\mathsf{E}[M_L] = \mathsf{E}[M_0] = 0$. Consequently, we have

$$\mathsf{E}[M_L] = \mathsf{E}[S_L - L \cdot w] = 0.$$

Next, on the one hand, $S_L \leq T$ trivially holds. On the other hand, since $c_j^{(s)} \leq j$ for

any epoch s, we have $S_L > T - j$ as otherwise L cannot be the last epoch. Therefore

$$T - j < \mathsf{E}[L] \cdot w \leq T,$$

and hence

$$\frac{T-j}{w} < \mathsf{E}[L] \leq \frac{T}{w}. \tag{4.3}$$

Finally, combining (4.2) and (4.3) yields

$$\mathsf{E}[r_j^{(1)}] \cdot \frac{T-j}{w} < \mathsf{E}\left[\sum_{s=1}^{L} r_j^{(s)}\right] = \mathsf{E}[r_j^{(1)}] \cdot \frac{T}{w},$$

implying the result. ∎

From the Theorem 1, it is clear that to maximize expected cumulative reward, we should devise an estimator for the expected per-round reward of each micro-arm. Let us introduce notation for the quantities we wish to estimate. In the following, let $(V, \tau) \sim Q_1$. For $j \in [D]$, define the expected per-round reward

$$g_{1,j} := \frac{\mathsf{E}\left[\mathbf{1}\left[j \geq \tau\right] \cdot V\right]}{\mathsf{E}\left[\min\{j, \tau\}\right]}. \tag{4.4}$$

A natural estimator of $g_{1,j}$ is

$$\hat{g}_{1,j}(s) := \frac{\sum_{m=1}^{s} \mathbf{1}\left[i_m = (1,j)\right] \cdot \mathbf{1}\left[j \geq \tau_m\right] \cdot V_m}{\sum_{m=1}^{s} \mathbf{1}\left[i_m = (1,j)\right] \cdot \min\{j, \tau_m\}}. \tag{4.5}$$

The above expression records the total reward received from pulls of arm-pair $(1, j)$ divided by the total rounds spent during these pulls. Comparing to Theorem 1, we can see that the numerator of (4.5) is an unbiased estimator of the numerator of Theorem 1, and the same relationship holds between the respective denominators.

However, the full estimator above is not an unbiased estimator of the expected per-round reward $g_{1,j}$, as is readily observed from Jensen's inequality. It is easy to see that (4.5) can easily be generalized to the case of multiple macro-arms as follows:

$$\hat{g}_{k,j}(s) := \frac{\sum_{m=1}^{s} \mathbf{1}\left[i_m = (k,j)\right] \cdot \mathbf{1}\left[j \geq \tau_{k,m}\right] \cdot V_m}{\sum_{m=1}^{s} \mathbf{1}\left[i_m = (k,j)\right] \cdot \min\{j, \tau_{k,m}\}}. \tag{4.6}$$

4.2 Wait-UCB

Our approach to solving F2A problems is to develop an upper confidence bound-style algorithm based on the reward per round estimate from (4.6). To achieve this, we develop a concentration inequality for how much higher the mean expected per-round reward may exceed this estimate. The following lemma gives us the concentration inequality for our quantity of interest.

Lemma 1 *Let X, Y be (possibly dependent) random variables with joint distribution P. Consider a sample $(X_1, Y_1), \ldots, (X_n, Y_n)$ of independent copies of $(X, Y) \sim P$. Assume that X takes values in $[0, 1]$ and Y takes values in $[1, B]$. Define $\mu_Y := \mathsf{E}[Y]$ and let $\hat{X}$ denote the sample mean of $X_1, \ldots, X_n$ (likewise for $\hat{Y}$ and $Y_1, \ldots, Y_n$). For any choice of $\delta \in [0, 1]$, we have with probability at least $1 - \delta$ over the sample,*

$$\left| \frac{\hat{X}}{\hat{Y}} - \frac{\mathsf{E}[X]}{\mathsf{E}[Y]} \right| \leq \sqrt{\frac{(B-1)\log\frac{4}{\delta}}{2n}} + \frac{2(B-1)\log\frac{4}{\delta}}{3\mu_Y n} + \sqrt{\frac{\log\frac{4}{\delta}}{2n}}.$$

Proof of Lemma 1 and an inversion corollary

Before proving Lemma 1, we first develop and prove a useful supporting lemma.

Lemma 2 *Let Y be a random variable taking values in $[1, B]$. Consider a sample $Y_1, \ldots, Y_n$ of independent copies of Y. Let $\hat{Y}$ denote the sample mean of $Y_1, \ldots, Y_n$*

and $\mu_Y := \mathsf{E}[Y]$. *Take* $\delta \in [0,1]$, *then with probability at least* $1-\delta$ *over the sample,*

$$\left|\frac{\hat{Y}-\mu_Y}{\mu_Y}\right| \leq \sqrt{\frac{(B-1)\log\frac{2}{\delta}}{2n}} + \frac{2(B-1)\log\frac{2}{\delta}}{3\mu_Y n}. \tag{4.7}$$

PROOF In order to get a tight concentration inequality for our problem, we will be applying Bernstein's inequality.

Fact 3 (Bernstein's Inequality (Bernstein, 1934)) *Assume that* $Z_1, \ldots, Z_n$ *are centered i.i.d. random variables satisfying* $|Z_j| \leq b$ *and* $\mathsf{Var}[Z_j] \leq \sigma^2$ *for all* $j \in [n]$. *Then*

$$\Pr\left(\frac{1}{n}\sum_{j=1}^{n} Z_j \geq t\right) \leq \exp\left(-\frac{nt^2}{2\left(\sigma^2 + \frac{b}{3}t\right)}\right).$$

Lemma 3 Bhatia-Davis Inequality (Bhatia and Davis, 2000) *Let* X *be a random variable bounded between* a *and* b *with* $\mathsf{E}[X] = \mu$. *Then,*

$$\mathsf{Var}[X] \leq (b-\mu)(\mu-a)$$ □

We begin by rewriting the left-hand side of (4.7) as follows:

$$\frac{\left|\hat{Y}-\mu_Y\right|}{\mu_Y} = \left|\frac{\hat{Y}}{\mu_Y} - 1\right|.$$

Next, observe that

$$\mathsf{Var}[Y] = \mathsf{E}[Y^2] - (\mathsf{E}[Y])^2 \leq (B-1)(\mu_Y - 1),$$

where we applied the Bhatia-Davis inequality and used the fact that $\mathsf{E}[|Y|] = \mathsf{E}[Y] = \mu_Y$. The last inequality may be coarse, but it turns out to simplify some things down

the road.

Taking $Z_j = Y_j - \mathsf{E}[Y_j]$, we may apply Bernstein's inequality with $b = B - 1$ and $\sigma^2 = (B - \mu_Y)(\mu_Y - 1)$, yielding

$$\begin{aligned}\Pr\left(\left|\frac{\hat{Y}}{\mu_Y} - 1\right| \geq t\right) &= \Pr\left(\left|\hat{Y} - \mu_Y\right| \geq \mu_Y t\right) \\ &\leq 2\exp\left(-\frac{n\mu_Y^2 t^2}{2((\mu_Y - 1)(B-1) + \frac{B-1}{3}\mu_Y t)}\right) \\ &= 2\exp\left(-\frac{nt^2}{2(\frac{(B-1)(\mu_Y-1)}{\mu_Y^2} + \frac{B-1}{3\mu_Y}t)}\right).\end{aligned}$$

Also, we can upper bound $\frac{(B-1)(\mu_Y-1)}{\mu_Y^2}$ by $\frac{(B-1)}{4}$, so that the above probability is at most

$$2\exp\left(-\frac{nt^2}{2(\frac{(B-1)}{4} + \frac{B-1}{3\mu_Y}t)}\right).$$

We can recover the discrepancy t by inverting the above equation (by setting the failure probability to δ), yielding

$$\log\left(\frac{2}{\delta}\right) = \frac{nt^2}{\frac{(B-1)}{2} + \frac{2(B-1)}{3\mu_Y}t} \quad .$$

Solving for t, we have that with probability at least $1 - \delta$,

$$t \leq \sqrt{\frac{(B-1)\log\frac{2}{\delta}}{2n}} + \frac{2(B-1)\log\frac{2}{\delta}}{3\mu_Y n}.$$

Therefore, as desired, with probability at least $1 - \delta$,

$$\left|\frac{\hat{Y} - \mu_Y}{\mu_Y}\right| \leq \sqrt{\frac{(B-1)\log\frac{2}{\delta}}{2n}} + \frac{2(B-1)\log\frac{2}{\delta}}{3\mu_Y n} \quad . \qquad \blacksquare$$

PROOF (OF LEMMA 1) Define $\mu_X := \mathsf{E}[X]$ and $\mu_Y := \mathsf{E}[Y]$. We begin with the rewrite

$$\frac{\mu_X}{\mu_Y} = \frac{\hat{X}}{\mu_Y} + \frac{\mu_X - \hat{X}}{\mu_Y} = \frac{\hat{X}}{\hat{Y}} + \left(\frac{\hat{X}}{\mu_Y} - \frac{\hat{X}}{\hat{Y}}\right) + \frac{\mu_X - \hat{X}}{\mu_Y}.$$

We bound the second and third terms in turn.

First, observe that

$$\left|\frac{\hat{X}}{\mu_Y} - \frac{\hat{X}}{\hat{Y}}\right| = \left|\hat{X}\left(\frac{\hat{Y} - \mu_y}{\mu_Y \hat{Y}}\right)\right| \leq \left|\frac{\hat{Y} - \mu_y}{\mu_y}\right|,$$

where the inequality is from the assumptions that $X \in [0,1]$ and $Y \geq 1$. Now, from Lemma 2, we have that with probability at least $1 - \delta$,

$$\left|\frac{\hat{Y} - \mu_Y}{\mu_Y}\right| \leq \sqrt{\frac{(B-1)\log\frac{2}{\delta}}{2n}} + \frac{2(B-1)\log\frac{2}{\delta}}{3\mu_Y n}.$$

Second, again using $Y \geq 1$, we have that

$$\left|\frac{\mu_X - \hat{X}}{\mu_Y}\right| \leq \left|\mu_X - \hat{X}\right|;$$

this term can be controlled using Hoeffding's inequality, where we now use that $X \in [0,1]$, this time yielding a deviation of size $\sqrt{\frac{\log\frac{2}{\delta}}{2n}}$. The result follows from a union bound. ∎

Next, we state a useful corollary of Lemma 1. The setup is identical to that of Lemma 1, but we restate the setup for the convenience of the reader. This corollary is simply an inversion of the aforementioned lemma.

Corollary 1 *Let X, Y be (possibly dependent) random variables with joint distribution P. Consider a sample $(X_1, Y_1), \ldots, (X_n, Y_n)$ of independent copies of $(X, Y) \sim$*

P. Assume that X takes values in $[0,1]$, and Y takes values in $[1,B]$. Define $\mu_Y := \mathsf{E}[Y]$ and let $\hat{X}$ denote the sample mean of $X_1,\ldots,X_n$ (likewise for $\hat{Y}$ and $Y_1,\ldots,Y_n$). For any $\epsilon > 0$, we have

$$\Pr\left(\left|\frac{\hat{X}}{\hat{Y}} - \frac{\mathsf{E}[X]}{\mathsf{E}[Y]}\right| \geq \epsilon\right) \leq 4\exp\left(-\left[\frac{-\frac{(\sqrt{B-1}+1)}{\sqrt{2n}} + \sqrt{\left(\frac{(\sqrt{B-1}+1)}{\sqrt{2n}}\right)^2 + \frac{8(B-1)\epsilon}{3n}}}{\frac{4(B-1)}{3n}}\right]^2\right) \tag{4.8}$$

The proof is by inversion and simply involves solving a quadratic equation.

With Lemmas 1 and (4.6) in hand, we now introduce the new algorithm, Algorithm 1. Since the arms correspond to waiting times, we call this algorithm "Wait-UCB".[1] From Lemma 1 and our later results in Section 4.3.1, for any arm pair (k,j) and non-negative integers s and n, the upper deviation around $\hat{g}_{k,j}(s)$ is given by

$$a_{k,j}(s) = \alpha_j \frac{\log s}{N_{k,j}(s)} + \beta_j \sqrt{\frac{\log s}{N_{k,j}(s)}},$$

where $\alpha_j := \frac{8(j-1)}{3}$ and $\beta_j := \sqrt{2}(\sqrt{j-1}+1)$ are constants independent of the macro-arms and $N_{k,j}(s)$ is the number of pulls of arm pair (k,j) at the end of epoch s.

Algorithm 1: Wait-UCB Algorithm

Input: time budget T, maximum waiting time D
$\alpha_j \leftarrow \frac{8(j-1)}{3}, \beta_j \leftarrow \sqrt{2}(\sqrt{j-1}+1)$ for all $j \in [D]$
$N_{k,j}(0) = 0$ for all $(k,j) \in [K] \times [D]$
$s = 1$
while $T > 0$ **do**
  Pull $i_s \leftarrow \underset{(k,j)\in[K]\times[D]}{\operatorname{argmax}} \hat{g}_{k,j}(s-1) + a_{k,j}(s-1)$ `// see (4.6) for` $\hat{g}_{k,j}(s-1)$
  $T \leftarrow T - \min\{j_s, \tau_s\}$, where $i_s = (k_s, j_s)$
  $N_{(k,j)}(s) \leftarrow N_{(k,j)}(s-1) + \mathbf{1}[i_s = (k,j)] \ \forall (k,j)$
  $s \leftarrow s+1$

Let us summarize the main idea behind the algorithm. Before an arm is pulled

[1] Also, "wait" is a homophone for "weight": the upper deviation terms are weighted by the waiting-time-dependent quantities α_j and β_j introduced below.

at least once, all the upper confidence bounds are initialized to infinity. This forces us to explore each arm at least once. After this phase, the arm that has the highest upper confidence bound is played. As an arm is played often, from the concentration inequality given in Lemma 1, our estimate of the ratio of reward to resource consumed by the arm becomes sharper and we converge to the optimal arm.

4.3 Expected regret of Wait-UCB

We begin bounding the regret of Wait-UCB by bounding the number of times a suboptimal arm is pulled.

4.3.1 Bound on expected number of pulls

Lemma 4 *For any sub-optimal arm pair (k, j): $\Delta_{k,j} := g_{k^*,j^*} - g_{k,j} > 0$, the expected number of pulls in L epochs is given by*

$$\mathsf{E}[N_{k,j}(L)] \leq \log(T) \left(\frac{\beta_j + \sqrt{\beta_j^2 + 2\alpha_j \Delta_{k,j}}}{\Delta_{k,j}} \right)^2 + \frac{4\pi^2}{3}.$$

Proof of Lemma 4

For convenience, we recall the upper deviation quantity defined earlier in the text which will be used in the proof. For all $(k, j) \in [K] \times [D]$ and any $s \geq 1$,

$$a_{k,j}(s) = \alpha_j \frac{\log s}{N_{k,j}(s)} + \beta_j \sqrt{\frac{\log s}{N_{k,j}(s)}}.$$

Proof (of Lemma 4) Let $g_{k,j}$ be the ratio of expected reward to expected waiting time for a pull of arm pair (k, j), and let (k^*, j^*) be the optimal arm pair, so that $g_{k^*,j^*} = \max_{(k,j)\in[K]\times[D]} g_{k,j}$. Let $N_{k,j}(s)$ be the number of pulls of arm pair (k, j)

until the end of epoch s. It will be useful to (implicitly) define a function $\hat{h}_{k,j}(\cdot)$ as $\hat{h}_{k,j}(N_{k,j}(s)) := \hat{g}_{k,j}(s)$; this function gives the empirical reward per round of arm pair (k,j) for $N_{k,j}(s)$ pulls. Let $u_{k,j}(N_{k,j}(s)) := a_{k,j}(s)$ be the confidence radius of arm (k,j) for $N_{k,j}(s)$ pulls. Let i_s denote the arm pair pulled in epoch s. Recall that L is the stopping time for the game. The number of pulls of suboptimal arm pair (k,j) with $\Delta_{k,j} > 0$ in L epochs is

$$N_{k,j}(L) = \sum_{s=1}^{L} \mathbf{1}\left[i_s = (k,j)\right]. \tag{4.9}$$

Since the time consumed in each epoch is stochastic (depending on the delay random variable $\tau_s \sim Q$), we first upper bound the random stopping time L by T; this is possible because each epoch lasts for at least one round. This upper bound, combined with the fact that each micro-arm is pulled once in the first D rounds, implies that (4.9) is at most

$$\sum_{s=1}^{T} \mathbf{1}\left[i_s = (k,j)\right] = 1 + \sum_{s=D+1}^{T} \mathbf{1}\left[i_s = (k,j)\right].$$

Let l be an arbitrary integer. We proceed by decomposing the second term into two sampling regimes. When $N_{k,j}(L) < l$, we say that the sub-optimal arm (k,j) is in the under-sampled regime and if $N_{k,j}(L) \geq l$, we say that the sub-optimal arm (k,j) is in the sufficiently sampled regime. The tuning of the value of l is given in detail in the proof of Lemma 5. Also, we show that when (k,j) is in the sufficiently sampled regime, we can use Corollary 1. Now, the summation in the RHS of the last line

above is equal to

$$\begin{aligned}
&\sum_{s=D+1}^{T} \mathbf{1}\left[i_s = (k,j), N_{k,j}(s-1) < l\right] + \sum_{s=D+1}^{T} \mathbf{1}\left[i_s = (k,j), N_{k,j}(s-1) \geq l\right] \\
&\leq l + \sum_{s=D+1}^{T} \mathbf{1}\left[i_s = (k,j); N_{k,j}(s-1) \geq l\right] \\
&\leq l + \sum_{s=D+1}^{T} \mathbf{1}\left[\hat{g}_{k,j}(s-1) + a_{k,j}(s-1) \geq \hat{g}_{k^*,j^*}(s-1) + a_{k^*,j^*}(s-1); N_{k,j}(s-1) \geq l\right] \\
&\leq l + \sum_{s=D+1}^{T} \mathbf{1}\left[\max_{l \leq p < s} \left\{\hat{h}_{k,j}(p) + u_{k,j}(p)\right\} \geq \min_{0 < m < s} \left\{\hat{h}_{k^*,j^*}(m) + u_{k^*,j^*}(m)\right\}\right] \\
&\leq l + \sum_{s=D+1}^{T} \sum_{m=1}^{s-1} \sum_{p=l}^{s-1} \mathbf{1}\left[\hat{h}_{k,j}(p) + u_{k,j}(p) \geq \hat{h}_{k^*,j^*}(m) + u_{k^*,j^*}(m)\right] \\
&\leq l + \sum_{s=1}^{T} \sum_{m=1}^{s-1} \sum_{p=l}^{s-1} \mathbf{1}\left[\hat{h}_{k,j}(p) + u_{k,j}(p) \geq \hat{h}_{k^*,j^*}(m) + u_{k^*,j^*}(m)\right]. \qquad (4.10)
\end{aligned}$$

Now, in (4.10), the inequality $\hat{h}_{k,j}(p) + u_{k,j}(p) \geq \hat{h}_{k^*,j^*}(m) + u_{k^*,j^*}(m)$ is possible only when at least one of the following three inequalities is true:

$$\hat{h}_{k^*,j^*}(m) + u_{k^*,j^*}(m) \leq g_{k^*,j^*}; \qquad (4.11)$$

$$\hat{h}_{k,j}(p) \geq g_{k,j} + u_{k,j}(p); \qquad (4.12)$$

$$g_{k^*,j^*} < g_{k,j} + 2u_{k,j}(p). \qquad (4.13)$$

Inequality (6) corresponds to $\hat{h}_{k^*,j^*}(m)$ being a significant underestimate of the optimal ratio g_{k^*,j^*}, while inequality (6) corresponds to $\hat{h}_{k,j}(p)$ being a significant overestimate of the suboptimal ratio $g_{k,j}$. Finally, inequality (4.13) will turn out to be false provided that l is selected to be large enough, as then, from $p \geq l$, the quantity $u_{k,j}(p)$ will be small enough to be strictly less than $\Delta_{k,j}/2$. Refer to the proof of Lemma 5

for the selection of l.

Taking the expectation on both sides of (4.10), we have

$$\begin{aligned}\mathsf{E}[N_{k,s}(L)] &\le l + \sum_{s=1}^{T}\sum_{m=1}^{s-1}\sum_{p=l}^{s-1}\Pr(\hat{h}_{k,j}(p) + u_{k,j}(p) \ge \hat{h}_{k^*,j^*}(m) + u_{k^*,j^*}(m)) \\ &\le l + \sum_{s=1}^{T}\sum_{m=1}^{s-1}\sum_{p=l}^{s-1}\Pr(\hat{h}_{k,j}(p) \ge g_{k,j} + u_{k,j}(p)) \\ &\qquad + \Pr(\hat{h}_{k^*,j^*}(m) \le g_{k^*,j^*} - u_{k^*,j^*}(m)). \qquad (4.14)\end{aligned}$$

We bound the above probabilities using Corollary 1; note that the time consumed by pulling an arm pair (k, j) will be at most j, so B in Corollary 1 becomes j and ϵ becomes $u_{k,j}(s) = \alpha_j \frac{\log s}{N_{k,j}(s)} + \beta_j \sqrt{\frac{\log s}{N_{k,j}(s)}}$.

For $\alpha_j = \frac{8(j-1)}{3}$, $\beta_j = \sqrt{2}(\sqrt{j-1}+1)$ and by setting
$l = \left\lceil \frac{\beta_j \sqrt{\log T} + \sqrt{\beta_j^2 \log T + 2\Delta_{k,j}\alpha_j \log T}}{\Delta_{k,j}} \right\rceil^2$, the above probabilities are bounded as

$$\begin{aligned}&\Pr(\hat{h}_{k,j}(p) \ge g_{k,j} + u_{k,j}(p)) \le 4\exp(-4\log s) = 4s^{-4}; \\ &\Pr(\hat{h}_{k^*,j^*}(m) \le g_{k,j^*} - u_{k^*,j^*}(m)) \le 4s^{-4}. \qquad (4.15)\end{aligned}$$

We explain in detail how we chose α_j, β_j, and l in the proof of Lemma 5.

Using the above bounds in (4.14), we have

$$\begin{aligned}
\mathsf{E}[N_{k,j}(L)] &\leq l + 8\sum_{s=1}^{T}\sum_{m=l}^{s-1}\sum_{p=l}^{s-1} s^{-4} \\
&\leq l + 8\sum_{s=1}^{T} s^{-2} \\
&\leq l + 8\sum_{s=1}^{\infty} s^{-2} \\
&\leq l + 8 \cdot \frac{\pi^2}{6} \\
&= l + 4 \cdot \frac{\pi^2}{3}
\end{aligned}$$

$$= \left(\frac{\sqrt{2}(\sqrt{j-1}+1)\sqrt{\log T}) + \sqrt{2(\sqrt{j-1}+1)^2 \log T + \frac{16}{3}\Delta_j (j-1)\log T}}{\Delta_j} \right)^2 + 4 \cdot \frac{\pi^2}{3},$$

where the last line is from Lemma 5, thus completing the proof. ∎

Lemma 5 *Let* $l = \log(T)\left(\frac{\beta_j + \sqrt{\beta_j^2 + 2\alpha_j \Delta_{k,j}}}{\Delta_{k,j}}\right)^2$. *Then for any epoch* s *such that* $l \leq s \leq L$,

$$\Pr\left(|g_{k,j} - \hat{g}_{k,j}(s)| \geq \epsilon\right) \leq 4s^{-4},$$

where $\epsilon = \alpha_j \frac{\log s}{N_{k,j}(s)} + \beta_j \sqrt{\frac{\log s}{N_{k,j}(s)}}$.

The proof of the above lemma is based on tuning the values of α_j and β_j in the algorithm. It follows by a number of sequence of algebraic steps and it is included in the Section 8.1 of additional proofs for brevity.

4.3.2 Regret bound

Theorem 4 *Wait-UCB's pseudo-regret is at most*

$$\sum_{(k,j)|\Delta_{k,j}>0} \mu_{k,j}^{(c)} \left(\frac{\left(\beta_j + \sqrt{\beta_j^2 + 2\Delta_{k,j}\alpha_j}\right)^2 \log T}{\Delta_{k,j}} + \mathcal{O}(1) \right),$$

where we define the mean waiting time $\mu_{k,j}^{(c)} := \mathsf{E}[c_{k,j}^{(1)}]$.

Proof The proof mainly uses the Lemma 4

$$\begin{aligned} R_T &= T \cdot g_{k^*,j^*} - \mathsf{E}\left[\sum_{s=1}^{L} r_{i_s}^{(s)}\right] + \mathcal{O}(1) \\ &\leq T \cdot g_{k^*,j^*} - \sum_{(k,j)} \mu_{k,j}^{(r)} \cdot \mathsf{E}[N_{k,j}(L)] + \mathcal{O}(1) \\ &= g_{k^*,j^*} \cdot (T - \sum_{(k,j)|\Delta_{k,j}=0} \mu_{k,j}^{(c)} \cdot \mathsf{E}[N_{k,j}(L)]) - \sum_{(k,j)|\Delta_{k,j}>0} \mu_{k,j}^{(r)} \cdot \mathsf{E}[N_{k,j}(L)]) \end{aligned} \tag{4.16}$$

Taking account of the random stopping time L, we can write,

$$T \leq \sum_{t=1}^{L} c_{i_s}^{(s)}.$$

Taking expectations on both sides, we get

$$\begin{aligned} T &\leq \sum_{k=1}^{K} \sum_{j=1}^{D} \mu_{k,j}^{(c)} \cdot \mathsf{E}[N_{k,j}(L)] \\ &= \sum_{(k,j)|\Delta_{k,j}=0} \mu_{k,j}^{(c)} \cdot \mathsf{E}[N_{k,j}(L)] + \sum_{(k,j)|\Delta_{k,j}>0} \mu_{k,j}^{(c)} \cdot \mathsf{E}[N_{k,j}(L)]. \end{aligned} \tag{4.17}$$

Substituting the above inequality (4.17) in the regret bound (4.16), we get

$$\begin{aligned} R_T &\leq g_{k^*,j^*} \cdot \left(\sum_{(k,j)|\Delta_{k,j}>0} \mu^{(c)}_{k,j} \cdot \mathsf{E}[N_{k,j}(L)] \right) - \sum_{(k,j)|\Delta_{k,j}>0} \mu^{(r)}_{k,j} \cdot \mathsf{E}[N_{k,j}(L)]) + \mathcal{O}(1) \\ &= \sum_{(k,j)|\Delta_{k,j}>0} \mu^{(c)}_{k,j} \left(g_{k^*,j^*} - g_{k,j} \right) \mathsf{E}[N_{k,j}(L)] + \mathcal{O}(1) \\ &= \sum_{(k,j)|\Delta_{k,j}>0} \mu^{(c)}_{k,j} \left(\Delta_{k,j} \right) \mathsf{E}[N_{k,j}(L)] + \mathcal{O}(1). \end{aligned}$$

From Lemma 4, we have an upper bound on $\mathsf{E}[N_{k,j}(L)]$, and so the regret R_T is at most

$$\sum_{(k,j)|\Delta_{k,j}>0} \mu^{(c)}_{k,j} \left(\frac{\left(\sqrt{2}(\sqrt{j-1}+1)\sqrt{\log T} \right) + \sqrt{2(\sqrt{j-1}+1)^2 \log T + \frac{16}{3}\Delta_{k,j}(j-1)\log T} \Big)^2}{\Delta_{k,j}} + 4(\frac{\pi^2}{3})\Delta_{k,j} \right) + \mathcal{O}(1)$$

The $\mathcal{O}(1)$ only hides moderate constants and scales as $\Delta_{k,j} \leq 1$. The leading term (involving $\log T$) in the above bound is less explicit due to the notation; a coarser version of the leading term is

$$\mathcal{O}\left(\sum_{(k,j)|\Delta_{k,j}>0} \mu^{(c)}_{k,j} \left(\frac{j \log T}{\Delta_{k,j}} \right) \right). \tag{4.18}$$

In comparing this result to previous regret bounds of Flajolet and Jaillet (2017), Xia et al. (2016), and Ding et al. (2013) for UCB-Simplex, Budget-UCB, and UCB-BV1 respectively, we focus on the case of a single macro-arm ($K = 1$), as this is enough to capture the "waiting" aspect of the problem. Let us assume that all suboptimal arms have gap lower bounded by $\Delta > 0$. Then since all the j are at most D, the leading

term (4.18) in our regret bound is of order at most

$$\left(\frac{D^2}{\Delta}\bar{\mu}^{(c)}\right)\log T, \tag{WAIT}$$

where we define $\bar{\mu}^{(c)} := \frac{1}{D}\sum_j \mu_{1,j}^{(c)}$. Now, in the worst case (when mean waiting times are high), this term becomes $(D^3/\Delta)\log T$. However, for easier problems where the mean waiting time for most arms is much smaller than D, the term improves to $(D^2/\Delta)\log T$. Before comparing this result to the regret bounds of Flajolet and Jaillet (2017), Xia et al. (2016), and Ding et al. (2013), it is important to note that those works have stochastic resource consumptions lying in $[0, 1]$. Therefore, to view the F2A framework in their setting, we rescale our consumptions from $[D]$ to $\{\frac{1}{D}, \frac{2}{D}, \ldots, 1\}$. Consequently, for each gap $\Delta_{1,j}$ in our paper, the corresponding gap in their paper will be scaled up by D.

With this conversion in mind, we turn our attention to Corollary 1 of Ding et al. (2013). After some unpacking, one can see that in an F2A problem, the leading term of their regret bound is of order

$$\left(D^4 + \frac{D^3}{\Delta} + \frac{D^2}{\Delta^2}\right)\frac{\mu_{1,j^*}^{(r)}}{\mu_{1,j^*}^{(c)}}\log T. \tag{DQZL}$$

In the situation where the optimal arm's mean waiting time $\mu_{1,j^*}^{(c)} \in [1, D]$ is small, their bound is noticeably worse: the first term is quartic in D, the second term matches our worst-case bound, and the third term is quadratic in $(1/\Delta)$. Yet, when the mean waiting time for the optimal arm is large, their bound becomes closer to the behavior of our bound were $\bar{\mu}^{(c)}$ to be small. In either case, their bound grows as $\frac{1}{\Delta^2}$.

Next, we compare our regret bound to a regret bound of Flajolet and Jaillet (2017) for UCB-Simplex. After converting their notation to ours, their Theorem 1

gives regret that is of order

$$\left(\frac{D^2}{\Delta}\sum_j \frac{1}{\mu_{1,j}^{(c)}}\right)\log T + \mathcal{O}(D^3). \tag{FJ}$$

The leading terms in (FJ) versus (WAIT) are close but there is an important distinction. Both leading terms contain a common factor of $\frac{D^2}{\Delta}$. However, (WAIT) has the average of the mean waiting times, while (FJ) has the sum of the reciprocal mean waiting times. When the mean waiting times are small, the average is smaller. When the mean waiting times are larger, the sum of reciprocals is smaller. Each quantity has a range of $[1, D]$. However, in any case, the constant term in (FJ) can be of order D^3, whereas the constant term from Theorem 4 (not shown but visible in the proof) is *always* $\mathcal{O}(D^2)$. A similar comparison can also be made for Budget-UCB from Theorem 3 of Xia et al. (2016) which gives the same logarithmic leading term as in (FJ) when the budget is sufficiently large.

We posit that the reason for our regret bound's improvement over the other bounds (in some regimes) is that our analysis is quite different: we directly form an upper confidence bound for the ratio estimator, and this gives us an opportunity to leverage a Bernstein-style improvement (from Bernstein's inequality). It would be interesting to somehow combine the style of analysis used in this work and the style from one of the other works to get a bound that dominates in all regimes. However, based on the experimental results in Section 5, it might be that our regret bounds for the *existing* Wait-UCB algorithm could be improved.

Chapter 5

Experiments

Our experiments focus on testing Wait-UCB in three different scenarios in the F2A framework. In all our experiments, we compare Wait-UCB to UCB-Simplex (Flajolet and Jaillet, 2017) and Budget-UCB (Xia et al., 2016); recall that these algorithms also admit logarithmic regret bounds in the BwK setting. Also, UCB-BV1 (Ding et al., 2013) was considered in all experiments, but it was later excluded as the other algorithms performed better. The delay τ_k and potential reward V_k are chosen such that they have a moderate minimum gap Δ. The delay distribution τ_k over $[D]$ for each experiment is given as a bar graph above the cumulative regret figures. Each experiment in this section is an average over 10 independent runs, and we use the same time budget of $T = 10^7$ rounds for all the experiments. The pseudo-regret in each experiment is calculated as

$$R_t = t \cdot g_{k^*,j^*} - \sum_{s=1}^{L_t} r_{i_s}^{(s)},$$

where L_t is the algorithm's last epoch for budget t.

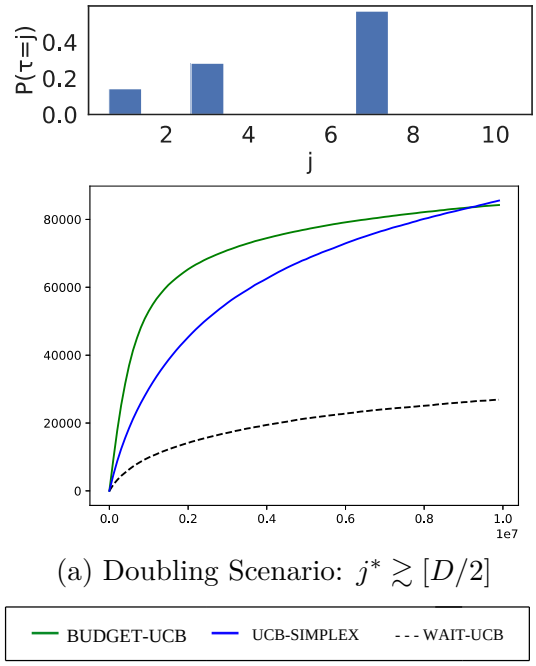
5.1 One macro-arm and several micro-arms

We start by having just 1 macro arm with deterministic potential reward $V_{k,s} = 1$ and $D = 10$ micro-arms configured with τ_k such that the optimal arm falls in different intervals of $[D]$, allowing us to understand the algorithms' behavior for various τ_k.

In the first experiment, we chose τ_k such that the delay doubles with the rewards obtained (see the bar graph in Figure 5.1a). For instance, if an algorithm chooses to wait for only 1 round, it consumes less resources and so can play for more epochs; on the flip side, if it decides to wait slightly longer than two rounds, it has twice the chance of getting the reward. This delay distribution induces a minimal gap of $\Delta = 0.042$. The learning algorithm must navigate a tight trade-off to select the optimal waiting time. In the next experiment, we chose the delay τ_k such that the optimal arm lies in the middle of $[D]$ incurring a moderate gap of $\Delta = 0.124$. The final experiment in this section stems from the observation that α_j becomes zero for arm $j = 1$, and so to study the behavior of the algorithm when the optimal arm is 1, an appropriate delay τ_k that incurs a moderate gap of $\Delta = 0.166$ is chosen.

Figure 5.3 shows the performance of Wait-UCB for the configurations of delay distribution that were discussed before. These experiments demonstrate a few interesting insights. From Figures 5.1a and 5.3a, we can see that for the first two experiments, Wait-UCB performs much better than UCB Simplex and Budget-UCB. We believe that the main reason for this comes from the underlying principle on which UCB algorithms are developed, i.e., Optimism in the Face of Uncertainty. In our F2A framework, the uncertainty in getting a reward decreases if we choose to wait for longer time. This behavior is very well captured by Wait-UCB in the construction of the confidence radius (which grows with j), resulting in relatively quick convergence towards the best arm. Also, observe that α_j becomes 0 for arm $j = 1$,

and in the last experiment (Figure 5.2a), Wait-UCB performs worse than the other algorithms; however, it still enjoys logarithmic regret.

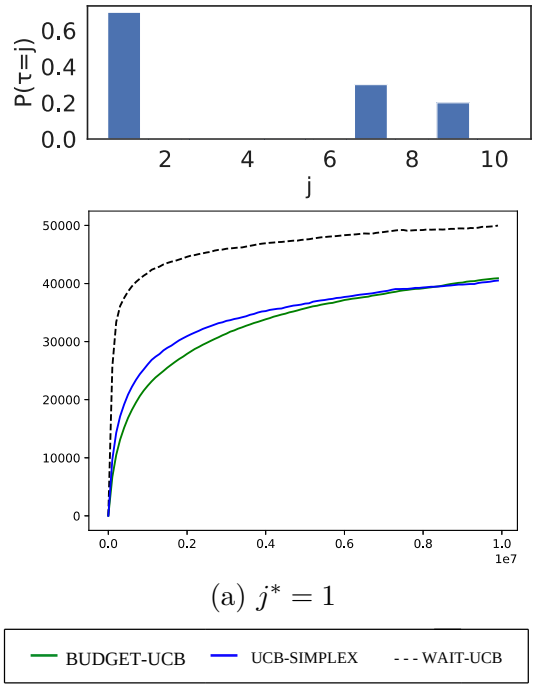


(a) Doubling Scenario: $j^* \gtrsim [D/2]$

Figure 5.1: Cumulative Regret of Wait-UCB for $D = 10, K = 1, V_k = 1$

5.2 Several macro-arms and one micro-arm

By having only one micro-arm ($D = 1$) and several macro-arms ($K > 1$), our problem reduces exactly to the standard MAB setting. Now, the algorithms simply need to pull the arm with highest V_k. A similar observation can be found in the setting when $D \geq 1$ and the delay distribution happens to be $P(\tau = 1) = 1$, i.e., any arm $j \in [D]$ will be the optimal arm as the epochs are completed in unit time (so the waiting time becomes insignificant here). Now, observe that the above two setups appear to be the same but are different in the perspective of algorithms which compute confidence radii for all the (k, j) pairs and select one among them. We decided to test the algorithms in the latter scenario with the following experimental setup. We took $K = 3$ and



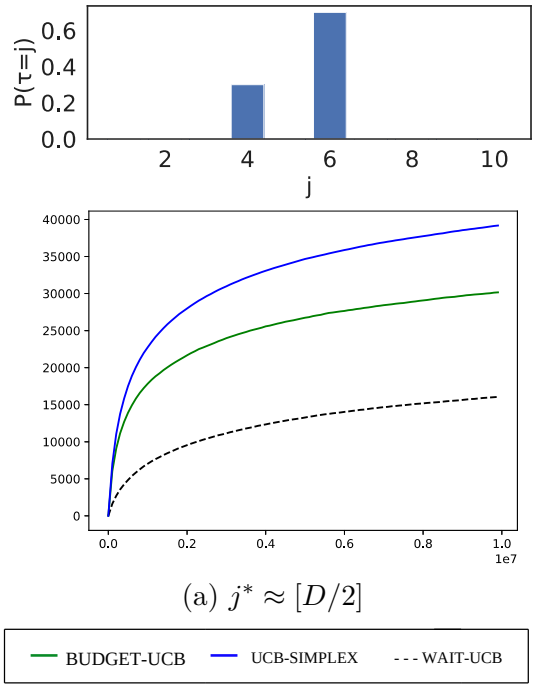
(a) $j^* = 1$

Figure 5.2: Cumulative Regret of Wait-UCB for $D = 10, K = 1, V_k = 1$

$D = 5$ with deterministic $\tau_k = 1$ and V_k being a Bernoulli random variable with success probabilities 0.5, 0.7, and 1 respectively.

Note that for this case, our upper confidence bound cleanly reduces to the upper confidence bound of UCB1 (Auer et al., 2002) for the standard stochastic multi-armed bandit problem, which is $a_{k,1}(s) = \sqrt{\frac{2 \log s}{N_{k,1}(s)}}$. Figure 5.4a shows the results for this setting. It is evident from the figure that Wait-UCB performs much better than UCB-Simplex and Budget-UCB.

We think this might be mainly due to the fact that the exploration term in Wait-UCB scales at most by the number of micro-arms j whereas, for UCB Simplex, the exploration term scales with the total number of Macro-Micro arm pairs $k \cdot D$. Due to this large exploratory factor, it takes longer for UCB-Simplex to converge.



(a) $j^* \approx [D/2]$

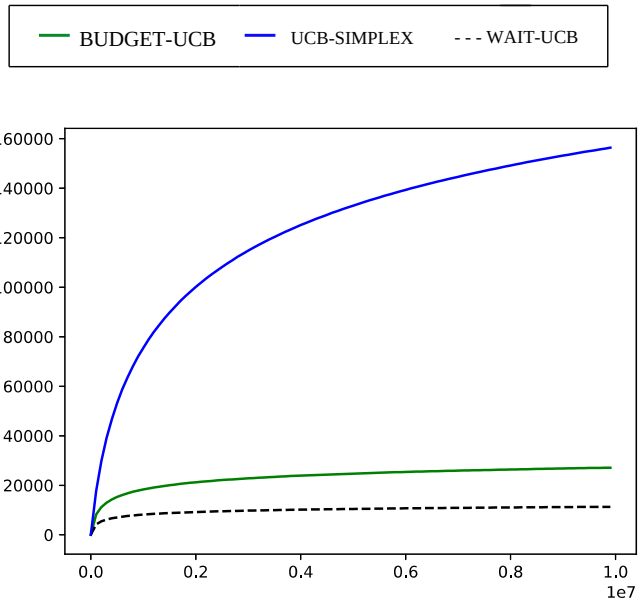
Figure 5.3: Cumulative Regret of Wait-UCB for $D = 10, K = 1, V_k = 1$

5.3 Several macro- and micro-arms

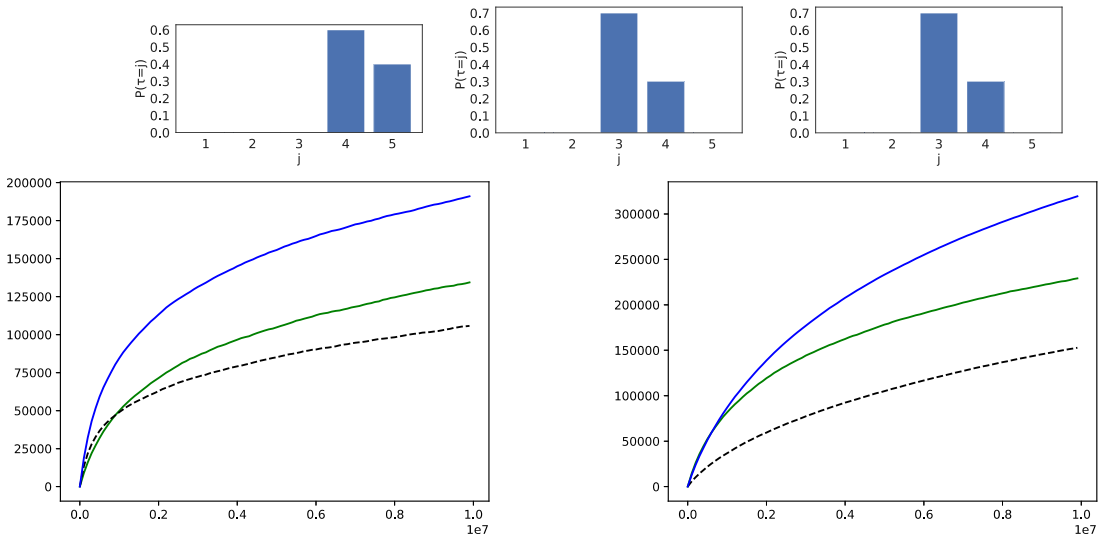
This setting is used in several of the real world applications discussed in Section 3.2. We performed a synthetic experiment inspired from computational advertising. We have a set of ads belonging to $K = 3$ categories with V_k their corresponding private utility for an impression (click) and $D = 5$ denotes the maximum waiting time for the ad before switching it. The delay distribution τ_k captures the time of impression of a user towards category k, which in a way reflects their interest on that category k. Note that showing a relevant ad affects the time of impression, thereby establishing an implicit connection between reward and delay. Now, the goal of the learning algorithm is to show the ad that best fits the interest of the user and learn the optimal waiting time for their impression.

We consider two cases for our experiments. In Case I, we set V_k to be a Bernoulli

random variable with success probabilities 0.7, 1, and 0.5 respectively. In Case II, we keep the same delay τ_k which represent the same user's interest but only change the V_k i.e., the publisher now receives different private utilities for different ad categories. The new Bernoulli random variable with success probabilities for the same categories are 1, 0.5, and 0.7 respectively. Figures 5.4b and 5.4c shows the learning behavior of Wait-UCB in these scenarios and the bar graphs on top represent the τ_k distribution for each of the 3 categories.



(a) for $D = 10, K = 1, V_k = 1$



(b) Case I: $D = 5, K = 3$

(c) Case II: $D = 5, K = 3$

Figure 5.4: Cumulative Regret of Wait-UCB

Chapter 6

Leftward Chain Feedback

Consider the crowdsourcing example discussed in the Chapter 1 where an agent has to dispatch tasks received from an employer to a pool of workers. Depending upon the efficiency of the selected worker, the time to complete the task varies. The agent dispatches the next task only when the current task is completed. If the selected worker is fast, it is worth waiting for the task to be completed. On the other hand, if the selected worker takes too long to complete the task, then it might be worth it to give up on the allocation of the task and reassign the same task to a new worker. The goal of the agent is to complete as many tasks as possible within a fixed time budget, and so for each task the agent selects a time and decides to wait for the task to be completed. If the task is not completed in the decided time, the agent gives up on that allocation of task and reassigns the task to a new worker from the pool. We can formalize this problem in the form of a repeated game. An interesting additional piece of information the agent gets during this process is that on top of just receiving the feedback for only the selected arm, it also gets to observe the reward signal for all the arms having lesser waiting time. That is if the agent has decided to wait for j units of time, then during the process of waiting, the agent can observe if the task is completed in any time lower than j. This helps the agent to not only observes

the reward for action j, but also to observe the rewards for all the times less than j. We call this additional feedback the agent receives as the "leftward chain" feedback.

For the rest of the chapter, we will consider only one macro-arm for notational simplicity as the technique and results can be extended to multiple macro arms. Let r_j and c_j be the actual reward obtained and cost incurred for pulling arm j with the delay random variable τ drawn from the distribution Q. Let $F_\tau(x)$ be the cumulative distribution function (CDF) of τ evaluated at x.

Figure 6.1 illustrates the leftward chain feedback in detail. The vertices of the graph represent the arms $j \in [D]$, which corresponds to the times the learning agent is willing to wait. When the agent pulls an arm, it observes the reward for all the arms connected in the directed feedback graph but only incurs the cost associated with pulling the arm that it pulled. In the F2A framework, the reward is spread across multiple arm in the form of a delay distribution τ compared to the standard multi-armed bandits problem, where each arm is associated with a separate reward distribution. Also, we can see that the feedback from any arm points in the direction to the arms on the left, hence the name.

This chapter is organized as follows: we first discuss in Section 6.1 on how we can model this feedback using a CDF, $F_\tau(j)$, without the use of an explicit graph. We then proceed to formulate an upper confidence bound around g_j in the form of $F_\tau(j)$ in Section 6.2. Using this, we develop a new algorithm Wait-2 Learn UCB in Section 6.3 and finally show that Wait-2 Learn UCB achieves a logarithmic problem-dependent regret bound, growing logarithmically in the number of rounds.

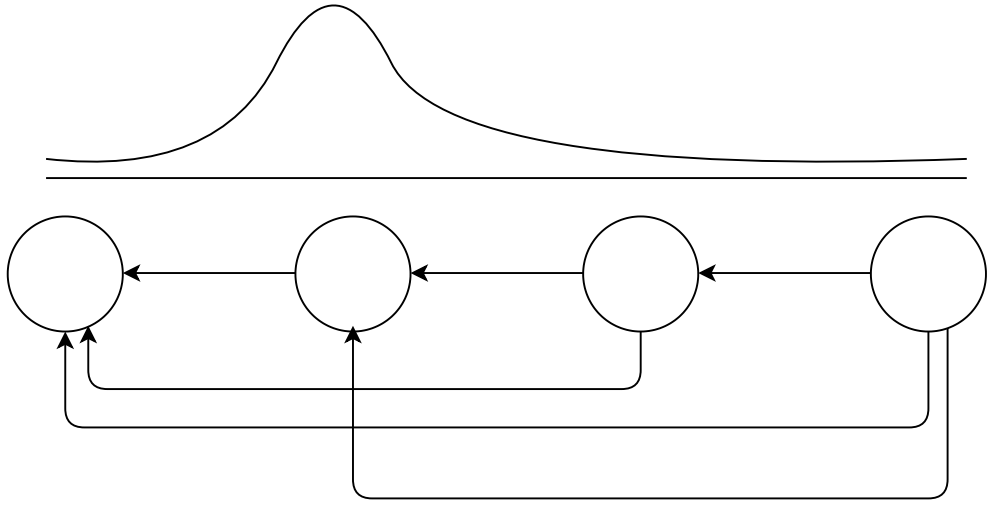


Figure 6.1: Leftward Chain Feedback, For $D = 4, K = 1$

6.1 Implicit Feedback on Delay τ

Let j be the action taken by the algorithm, i.e., the arm pulled in some arbitrary epoch. For now, let us consider only one macro arm, $K = 1$ and the private utility of the reward to be a unit quantity, $V_k = 1 \ \ \forall k \in [K]$, and so we can write the reward collected r_j and the resource consumed c_j as follows,

$$r_j = \mathbf{1}\left[j \geq \tau\right] \tag{6.1}$$

$$c_j = \min(j, \tau). \tag{6.2}$$

Recall from Chapter 2 that for the F2A framework, the right metric to capture the problem setting is g_j, which is the ratio of expected reward of the arm to the expected cost, i.e., the expected reward per unit time as opposed to just the reward observed in each epoch. We can write g_j as,

$$g_j = \frac{\mathsf{E}[r_j]}{\mathsf{E}[c_j]}. \tag{6.3}$$

Fact 5 (Expectation of a non-negative RV - Complementary CDF) *Let X be a random variable that takes on only non-negative integer values. Then the expectation of X can be written as*

$$\mathsf{E}[X] = \sum_{i=0}^{\infty} \Pr(X > i) = \sum_{i=1}^{\infty} \Pr(X \geq i).$$

Let us now look at the individual quantities of (6.3) in detail, both the expectation of the reward and the cost from (6.1) and (6.2) respectively. Let us first look at the expectation of the resource consumed. Since the resource consumed is always a positive value, from Fact 5 we can write it as the following:

$$\begin{aligned}
\mathsf{E}[c_j] &= \sum_{i=1}^{\infty} \Pr(c_j \geq i) \\
&= \sum_{i=1}^{\infty} \Pr(j \geq i; \tau \geq i) \\
&= \sum_{i=1}^{\infty} \mathbf{1}[i \leq j] \Pr(\tau \geq i) \\
&= \sum_{i=1}^{j} \Pr(\tau \geq i) \\
&= \sum_{i=1}^{j} (1 - \Pr(\tau < i)).
\end{aligned}$$

We can write the above expression in terms of the CDF of the delay τ as

$$\mathsf{E}[c_j] = \sum_{i=1}^{j} (1 - F_\tau(i-1)). \tag{6.4}$$

Similarly, $\mathsf{E}[r_j]$ can also be written in terms of the CDF of the delay τ as

$$\begin{aligned}
E[r_j] &= P(j \geq \tau) \\
&= F_\tau(j).
\end{aligned} \tag{6.5}$$

From (6.5) and (6.4), we can write the g value for arm j and delay τ as

$$g_j = \frac{\mathsf{E}[r_j]}{\mathsf{E}[c_j]} = \frac{\Pr(j \geq \tau)}{\mathsf{E}[\min\{j, \tau\}]} = \frac{F_\tau(j)}{\sum_{i=1}^{j}[1 - F_\tau(i-1)]}. \tag{6.6}$$

Now, we can see that g_j is expressed as a function of the CDF of the delay, F_τ. This estimate takes full use of observing the rewards according to the leftward feedback chain, enabling us to implicitly use this information without explicitly using a feedback graph.

Our estimator of the above mentioned g_j is

$$\hat{g}_j = \frac{\hat{F}_\tau(j)}{\sum_{i=1}^{j}[1 - \hat{F}_\tau(i-1)]},$$

where, $\hat{F}_\tau(x) = \dfrac{\sum_{q=1}^{s} \mathbf{1}\left[I_s \geq x; \tau \leq x\right]}{\sum_{q=1}^{s} \mathbf{1}\left[I_s \geq x\right]}$ is the CDF estimate for pulling arm I_s

at epoch s with τ being the delay distribution.

Note that we have ignored the epoch number in the notation of the estimates $\hat{g}_j$ and $\hat{F}_\tau(x)$ as they both corresponds for epoch s throughout this chapter and are avoided for notational brevity.

6.2 UCB in $F_\tau(j)$

Now we can see that the estimate $\hat{g}_j$ for every epoch, can also utilize the rewards of the arms having lesser waiting time than arm j that is being pulled. The next essential step is to construct an upper confidence radius around the quantity $\hat{g}_j$. This

will enable us to design an algorithm that works on the basis of "optimism in the face of uncertainty". For that, it is essential to first look at the upper confidence radius around $F_\tau(j)$.

Theorem 6 (DKW Inequality) Dvoretzky et al. (1956) *At a time t, let $\hat{F}_n(x)$ be the empirical distribution function of $F(x)$. The probability that the maximum of the difference between $\hat{F}_n(x)$ and $F(x)$ over all x with n samples is at least ϵ is less than $2e^{-2n\epsilon^2}$, i.e.,*

$$P\left(\sup_x |\hat{F}_n(x) - F(x)| > \epsilon\right) \leq 2e^{-2n\epsilon^2}.$$

Let ϵ_j be the upper confidence radius around $\hat{F}_\tau(j)$. Then, using the above theorem in our setting, we get the following:

With probability at least $1-\delta$ over the samples for $\delta \in [0,1]$, the deviation between the true CDF of the delay and its estimate at epoch s is,

$$|\hat{F}_\tau(j) - F_\tau(j)| \leq \epsilon_j \ \forall j \in [D]. \tag{6.7}$$

Throughout this chapter, $\epsilon_j = \sqrt{\frac{2\log(1/\delta)}{N_j(s)}}$ is the upper deviation around $\hat{F}_\tau(j)$ for arm j at epoch s, where $N_j(s)$ is the number of samples for arm j and $\hat{F}_\tau(j)$ denote the estimate of the CDF of τ for arm j at epoch s.

Let r_j be the true expected reward for pulling arm j and $\hat{r}_j$ be the estimate of the expected reward. From (6.5) we know that,

$$r_j = \Pr(j \geq \tau) = F_\tau(j).$$

From the above equation and (6.7), we can say that with probability at least $1-\delta$, the following holds true,

$$r_j \in [\hat{r}_j - \epsilon_j, \hat{r}_j + \epsilon_j] \;\; \forall j \in [D]. \tag{6.8}$$

Similarly, let c_j be the expected cost for pulling arm j and $\hat{c}_j$ be the corresponding empirical estimate.

Lemma 6 *With probability at least $1-\delta$, the estimated cost $\hat{c}_j$ for pulling an arm j in F2A problem is within a deviation of $j\epsilon_{j-1}$ from the true cost c_j, i.e.,*

$$|c_j - \hat{c}_j| \leq j\epsilon_{j-1} \;\forall j \in [D]. \qquad \square$$

From (6.4), the expected cost of pulling the arm j is

$$c_j = \mathsf{E}[\min\{j, \tau\}] = \sum_{i=1}^{j} \left[1 - F_\tau(i-1)\right].$$

Now, let us calculate the confidence radius for the difference between $\hat{c}_j$ and c_j. From the above equation we can write that

$$\begin{aligned}
|c_j - \hat{c}_j| &= \left| \sum_{i=1}^{j} \left[1 - F_\tau(i-1)\right] - \sum_{i=1}^{j} \left[1 - \hat{F}_\tau(i-1)\right] \right| \\
&= \left| \left[j - \sum_{i=1}^{j} F_\tau(i-1) \right] - \left[j - \sum_{i=1}^{j} \hat{F}_\tau(i-1) \right] \right| \\
&= \left| \left[\sum_{i=1}^{j} \hat{F}_\tau(i-1) \right] - \left[\sum_{i=1}^{j} F_\tau(i-1) \right] \right| \\
&= \left| \left[\sum_{i=1}^{j} \hat{F}_\tau(i-1) - \sum_{i=1}^{j} F_\tau(i-1) \right] \right|.
\end{aligned}$$

From (6.7) the above equation can be upper bounded as

$$\sum_{i=1}^{j} \epsilon_{i-1}.$$

Upper bounding the above expression, we get

$$j\epsilon_{j-1}.$$

The last inequality is due to the fact that $\epsilon_j \geq \epsilon_{j-1}$ as ϵ_j is the discrepancy between the true CDF evaluated at j, $F_\tau(j)$, and its estimate $\hat{F}_\tau(j)$ and it is non-decreasing in j.

Thus,

$$|c_j - \hat{c}_j| \leq j\epsilon_{j-1}.$$

From this we can say that with probability at least $1 - \delta$,

$$c_j \in [\hat{c}_j - j\epsilon_{j-1}, \hat{c}_j + j\epsilon_{j-1}]\,. \tag{6.9}$$

6.2.1 UCB for g_j

From the previous chapters, we know that to construct a UCB style algorithm, we need to construct a confidence radius around the quantity g_j for our problem. To utilize the leftward chain feedback implicitly, we need to express both the g_j and the confidence radius in terms of the CDF. In the previous sections, we have already established that both the reward and cost can be expressed in terms of the CDF, F_τ. With the available information, let us now focus on the actual UCB around g_j.

Lemma 7 *With probability at least* $1-\delta$*, the confidence radius for the ratio* $\hat{g}_j$ *is given as*

$$|\hat{g}_j - g_j| \leq \frac{j\epsilon_j + \sum_{i=1}^{j} \min\{\hat{F}_\tau(j) + \epsilon_j, 1\}\,[\epsilon_j] + [j\epsilon_{j-1}]\min\{\hat{F}_\tau(j) + \epsilon_j, 1\}}{j^2 - j\left(\sum_{i=1}^{j} \hat{F}_\tau(i-1) + [j\epsilon_{j-1}]\right) - j\sum_{i=1}^{j} \hat{F}_\tau(i-1) + \sum_{i=1}^{j} \hat{F}_\tau(i-1) \sum_{i=1}^{j} \max\{\hat{F}_\tau(i-1) - \epsilon_{i-1}, 0\}}$$

□

The RHS of the above expression is defined as $a_j(s)$, the upper deviation around $\hat{g}_j$ for $N_j(s)$ number of pulls.

The proof of this lemma comprises of a sequence of algebraic steps to arrive at a quantity that is observable and can be used by the algorithm. For a detailed proof, refer to Section 8.2 in the additional proofs.

6.3 Wait-2 Learn UCB Algorithm

Let us see the overall working of our UCB style algorithm. Before the start of the first epoch, the upper confidence bounds of all the arms are initialized to infinity. In each epoch, the algorithm pulls the arm which has the highest upper confidence bound. Since the upper confidence bound of all the arms are infinity at the start, this ensures that all the arms are played at least once. Once the algorithm starts playing in this fashion, the upper confidence bound of the estimate of ratio of reward to resource consumed by the arm becomes narrower and the estimate $\hat{g}_j$ comes closer to the expected value of the ratio of reward to resource consumed of all the arms. Thus the algorithm progresses towards pulling the optimal arm to maximize the total collected reward.

Algorithm 2: Wait-2 Learn UCB

Input: time budget T, maximum waiting time D
$N_j(0) = 0$ for all $j \in [D]$
$s = 1$
while $T > 0$ **do**
  Pull $i_s \leftarrow \underset{(j)\in[D]}{\operatorname{argmax}}\, \hat{g}_j(s-1) + a_j(s-1)$
  // see Lemma 7 for $a_j(s-1)$
  $T \leftarrow T - \min\{j_s, \tau_s\}$, where $i_s = j_s$
  $N_j(s) \leftarrow N_j(s-1) + \mathbf{1}\left[i_s = j\right]\ \forall j \in D$
  $s \leftarrow s + 1$

Wait-2 Learn UCB is an updated version of Wait-UCB with the ability to handle the extra feedback, and since the learner waits in each epoch to learn about the delay distribution we have given the name Wait-2 Learn UCB to our algorithm.

6.4 Expected Regret of Wait-2 Learn UCB

Let Δ_j be the gap between the ratio g for the best arm j^* and arm j. We begin bounding the regret of Wait-2 Learn UCB by bounding the number of times a suboptimal arm is pulled.

6.4.1 Expected Number of sub-optimal pulls

Recall from previous chapters that L is the random stopping time, which is the last epoch (the epoch at which the budget T runs out).

Lemma 8 *For any sub-optimal arm j: $\Delta_j := g_{j^*} - g_j > 0$, the expected number of pulls in L epochs, is given by*

$$\mathsf{E}[N_j(L)] \leq \frac{8j^2}{\Delta^2}\log T + 2\left(1 + \exp\left(-j^2\right)\right)\frac{\pi^2}{3}.$$

The expected number of sub-optimal pulls for Wait-2 Learn UCB can be bounded in similar fashion as Wait-UCB.

PROOF (OF LEMMA 8) We follow a similar proof strategy as Lemma 4. Let i_s be the arm pulled in epoch s, L be the stopping time which is the epoch at which the budget T runs out and D be the maximum number of arms. Let us consider an arbitrary integer l for decomposing the sampling regime. When the sub-optimal arm is in the under-sampled regime we denote by $N_j(s) < l$ and $N_j(s) \geq l$ is used to denote when the sub-optimal arm is in sufficiently sampled regime. It will be useful to define new functions to represent the estimates in terms of the number of samples. Let $\bar{r}_{j,N_j(s)} := \hat{r}_j(s)$ and $\bar{c}_{j,N_j(s)} := \hat{c}_j(s)$ be the empirical reward and the empirical cost of arm j for $N_j(s)$ pulls. Similarly let $\hat{F}_{\tau,N_j(s)}(j)$ be the empirical estimate of the CDF of τ for arm j with $N_j(s)$ pulls. With this information, let us proceed with bounding the number of pulls of sub-optimal arms. We have

$$\begin{aligned}
N_j(L) &= \sum_{s=D+1}^{T} \mathbf{1}\left[i_s = j, N_j(s-1) < l\right] + \sum_{s=D+1}^{T} \mathbf{1}\left[i_s = j, N_j(s-1) \geq l\right] \\
&\leq l + \sum_{s=D+1}^{T} \mathbf{1}\left[i_s = j; N_j(s-1) \geq l\right] \\
&\leq l + \sum_{s=D+1}^{T} \mathbf{1}[\hat{g}_j(s-1) + a_j(s-1) \geq \\
&\qquad\qquad \hat{g}_{j^*}(s-1) + a_{j^*}(s-1); N_j(s-1) \geq l].
\end{aligned}$$

Taking the expectation on both side, we get

$$\mathsf{E}[N_j(L)] \leq l + \sum_{s=D+1}^{T} \Pr\{\hat{g}_j(s-1) + a_j(s-1) \geq \hat{g}_{j^*}(s-1) + a_{j^*}(s-1); N_j(s-1) \geq l\}.$$

The above inequality is possible only when at least one of the following three events are

true:

$$\hat{g}_j(s) \geq g_j + a_j(s) \tag{6.10}$$

$$\hat{g}_{j^*}(s) \leq g_{j^*} - a_j(s) \tag{6.11}$$

$$g_{j^*} < g_j + 2a_j(s) \tag{6.12}$$

Event I corresponds to overestimating g_j as shown in (6.10). Event II corresponds to underestimating g_{j^*} as shown in (6.11). Finally for both Event I and Event II to be true, equation (6.12) should be satisfied (Event III). Let us look at each of the events separately in detail.

Event I: Overestimating g_j

From (6.8) and Lemma 6, we can write the following,

$$\mathsf{E}\left[\sum_{s=1}^{L} \mathbf{1}\left[\hat{g}_j(s) \geq g_j + a_j(s)\right]\right] \leq \sum_{s=1}^{\infty} \Pr[\hat{r}_j(s) \geq r_j + \epsilon_j] + \Pr[\hat{c}_j(s) \leq c_j - j\epsilon_{j-1}].$$

From Theorem 6, we can use the value of the ϵ_j and the above quantity is upper bounded

by

$$
\begin{aligned}
&\sum_{s=1}^{\infty}\sum_{p=1}^{s}\Pr\left[\bar{r}_{j,p}\geq r_j+\sqrt{\frac{2\log(s)}{p}}\right]\\
&+\Pr\left[\bar{c}_{j,p}\leq c_j-j\sqrt{\frac{2\log(s)}{p}}\right]\\
&\leq\sum_{s=1}^{\infty}\sum_{p=1}^{s}\Pr\left[\hat{F}_{\tau,p}(j)\geq F_\tau(j)+\sqrt{\frac{2\log(s)}{p}}\right]\\
&+\Pr\left[\bar{c}_{j,p}\leq c_j-j\sqrt{\frac{2\log(s)}{p}}\right]\\
&\leq\sum_{s=1}^{\infty}\sum_{p=1}^{s}2\exp\left(-4\log s\right)+2\exp\left(-4j^2\log s\right)\\
&\leq\sum_{s=1}^{\infty}\sum_{p=1}^{s}2\exp\left(-4\log s\right)+2\exp\left(-4\log s\right)\cdot\exp\left(-j^2\right)\\
&\leq\sum_{s=1}^{\infty}\sum_{p=1}^{s}2\exp\left(-4\log s\right)\left(1+\exp\left(-j^2\right)\right)\\
&\leq\left(1+\exp\left(-j^2\right)\right)\sum_{s=1}^{\infty}\sum_{p=1}^{s}2\exp\left(-4\log s\right)\\
&\leq\left(1+\exp\left(-j^2\right)\right)\frac{\pi^2}{3}.
\end{aligned}
$$

Event II: Underestimating g_{j^*}

We follow a similar bounding strategy as Event I and Event II is bounded by,

$$
\mathsf{E}\left[\sum_{s=1}^{L}\mathbf{1}\left[\hat{g}_{j^*}(s)\leq g_{j^*}-a_j(s)\right]\right]\leq\left(1+\exp\left(-j^2\right)\right)\frac{\pi^2}{3}.
$$

Event III: Sufficiently sampled

Finally, inequality (6.12) will turn out to be false provided that l is selected to be large enough, as then, from $p\geq l$, the quantity $a_j(s)$ will be small enough to be strictly less than $\Delta_j/2$. Refer to the proof of Lemma 5 for the selection of l. For this value of $l=\frac{8j^2}{\Delta_j^2}\log s$, the condition $a_j(s)\leq\frac{\Delta_j}{2}$ holds True.

Combining all the above 3 events, we can bound the expected number of sub-optimal arm pulls

$$\mathsf{E}[N_j(L)] \leq \frac{8j^2}{\Delta_j^2} \log s + 2\left(1 + \exp\left(-j^2\right)\right) \frac{\pi^2}{3}.$$

■

6.5 Regret Bound

Theorem 7 *Wait-2 Learn UCB's pseudo-regret is at most*

$$R_T = \sum_{j|\Delta_j>0} \mu_j^{(c)} \frac{8j^2}{\Delta_j} \log T + 2\left(1 + \exp\left(-j^2\right)\right) \frac{\pi^2}{3} \mu_j^{(c)} + \mathcal{O}(1),$$

where we the mean waiting time $\mu_j^{(c)} := \mathsf{E}[c_j]$.

PROOF The proof of Theorem 7 mainly uses the results from Lemma 8.

Recalling the regret definition from Chapter 2, for the sake of simplicity let us consider the case of one macro arm, yielding

$$R_T = T \cdot g_{j^*} - \mathsf{E}\left[\sum_{s=1}^{s=L} r_{i_s^{(s)}}\right].$$

The regret shown in the previous expression can be expressed in terms of the number of pulls of the suboptimal arms and the gap as

$$R_T = \sum_{j|\Delta_j>0} \mu_j^{(c)} \left(\Delta_j\right) \mathsf{E}[N_j(L)] + \mathcal{O}(1). \tag{6.13}$$

Now, let us use the results from Lemma 8 to bound the expected number of pulls of

any suboptimal arm. With that, we can upper bound the above equation as follows:

$$\sum_{j|\Delta_j>0} \mu_j^{(c)}\left(\Delta_j\right)\left(\frac{8j^2}{\Delta_j^2}\log T + 2\left(1+\exp\left(-j^2\right)\right)\frac{\pi^2}{3}\right) + \mathcal{O}(1).$$

Thus, we can write the pseudo-regret of Wait-2 Learn UCB as

$$R_T \leq \sum_{j|\Delta_j>0} \mu_j^{(c)}\frac{8j^2}{\Delta_j}\log T + 2\left(1+\exp\left(-j^2\right)\right)\frac{\pi^2}{3}\mu_j^{(c)}\Delta_j + \mathcal{O}(1). \tag{6.14}$$

■

We can write the above regret bound in a concise form in terms of D and T, in which case the above (6.14) becomes $\mathcal{O}\left(\sum_{(j)|\Delta_j>0}\frac{D^3}{\Delta_j}\log T\right)$. We can clearly see that the regret bound of Wait-UCB is better than that of Wait-2 Learn UCB. The best case scenario happens when the last arm $j = D$ (the arm which enables the algorithm to observe the rewards of all the other arms) gets pulled repeatedly until it is sufficiently sampled first. In that case, all the remaining arms also get at least the same number of sufficient samples because of the leftward-chain feedback. Then, the regret bound of Wait-2 Learn UCB reduces to only $\mathcal{O}\left(\frac{D^3}{\Delta_j}\right)$, thus matching the regret bound of the Wait-UCB only in this best case scenario. However, the main contribution of this work also lies in addressing this leftward-chain feedback without the explicit use of a directed feedback graph. The regret bound could be further improved if we were to use an elimination style algorithm similar to Cohen et al. (2016) or FiringUCB (Jain and Jamieson, 2018). The latter uses a UCB style approach to search over the policy class by splitting them into brackets defined in terms of the number of flips for a coin. In each round, the bracket with the highest UCB is pulled and the learning agent accordingly allocates the number of flips to see if the drawn coin is fired or not and then iteratively performs the elimination. Though the problem setting cannot be mapped directly with the F2A problem, one can try to adopt a similar strategy

where we split the waiting times into brackets and proceed as follows. In each epoch, the learning agent pulls the highest indexed arm in the bracket with the largest UCB to enable the learner to also observe additional feedback. Then a similar elimination-style approach can be taken to facilitate a reduction in the arm set. This way, we should be able to further reduce the regret's dependence on the problem parameter D.

Chapter 7

Conclusion and Future work

In this work, we introduced the Farewell to Arms framework of multi-armed bandit problems and presented a new algorithm, Wait-UCB, along with a logarithmic problem-dependent regret bound of order $O((D^3/\Delta)\log T)$. In the case of a single macro-arm and when most micro-arms have low mean waiting times (much smaller than D), our regret bound improves to $O((D^2/\Delta)\log T)$; in this regime, the leading $(\log T)$ term of our bound for Wait-UCB is smaller by a factor of D than the bounds of Ding et al. (2013), Xia et al. (2016), and Flajolet and Jaillet (2017) for UCB-BV1, Budget-UCB, and UCB-Simplex respectively. However, in the opposite regime, the leading term (but not the constant term) of the bound for UCB-Simplex and Budget-UCB can be smaller than ours. Yet, our experiments show that Wait-UCB empirically outperforms UCB-Simplex and Budget-UCB in a number of settings. Notably, in the standard bandit setting with only one arm, our algorithm collapses to be equal to a standard UCB algorithm, while this is not the case for the other algorithms. Indeed, our algorithm well-outperforms UCB-Simplex and Budget-UCB even in this simple setting. Also, a new algorithm, Wait-2 Learn UCB, for the case with additional leftward feedback was also developed which exhibits a logarithmic problem dependent regret bound of order $\mathcal{O}\left(\sum_{j|\Delta_j>0}\frac{D^3}{\Delta}\log T\right)$ which matched with the regret bound of

Wait-UCB in the best case scenario, which happens when the last arm D is repeatedly pulled and sufficiently sampled.

In closing, we mention a few exciting directions for future work. First, It would be interesting to explore more along the direction of Wait-2 Learn UCB for the leftward chain feedback and potentially gain improvements in the regret bound over Wait-UCB. For example, if a learning agent decides to wait for 5 rounds, the learning agent also gains feedback for any shorter waiting time. This feedback could readily be used by Wait-UCB, but proving improved regret bounds in light of this feedback is highly challenging. A second, further improvement would be to completely utilize the structure of the F2A game, which also includes conditional feedback. For instance, observe that if the learning algorithm decides to wait for $j \geq \tau$ rounds (for delay τ), then the algorithm knows τ and hence also gets feedback for all the longer waiting times $j+1, \ldots, D$. Yet, if it does not wait for long enough (i.e., $j < \tau$), then this feedback will not be available.

Chapter 8

Additional Proofs

8.1 Proof of Lemma 5

In the following proof, since j is fixed throughout, we simply write α and β instead of α_j and β_j (i.e. we drop the subscripts).

Proof We now find a value of l that marks the sufficient sampling regime (this regime was described in the proof of Lemma 4). We know that inequality (4.13) is only true for $p < l$. We will use this argument to find the lower bound on l. Note that the value of l is different for different pairs of arms; we used l for notation consistency with p. The following three inequalities are equivalent:

$$\begin{aligned} u_{k,j}(p) &< \frac{\Delta_{k,j}}{2}; \\ \alpha\frac{\log s}{p} + \beta\sqrt{\frac{\log s}{p}} &< \frac{\Delta_{k,j}}{2}; \\ \alpha \log s + \beta\sqrt{p \log s} &< p\frac{\Delta_{k,j}}{2}. \end{aligned}$$

The last line above is quadratic in $Z = \sqrt{p}$. Take $a = \frac{-\Delta_{k,j}}{2}$, $b = \beta\sqrt{\log s}$, and

$c = \alpha \log s$ to be the coefficients for the quadratic form. Then we see that

$$Z < \frac{-\beta\sqrt{\log s} - \sqrt{\beta^2 \log s + 2\Delta_{k,j}\alpha \log s}}{-\Delta_{k,j}}$$

and hence

$$p < \left[\frac{\beta\sqrt{\log s} + \sqrt{\beta^2 \log s + 2\Delta_{k,j}\alpha \log s}}{\Delta_{k,j}}\right]^2,$$

so, for $p \geq \left[\frac{\beta\sqrt{\log s}+\sqrt{\beta^2 \log s+2\Delta_{k,j}\alpha \log s}}{\Delta_{k,j}}\right]^2$, inequality (4.13) is false. Therefore,

$$l \geq \left[\frac{\beta\sqrt{\log T} + \sqrt{\beta^2 \log T + 2\Delta_{k,j}\alpha \log T}}{\Delta_{k,j}}\right]^2.$$

To get the values of α_j and β_j to bound the probabilities as in (4.15), we start by equating the probability quantity obtained from Corollary 1 to $4\exp(-4\log s)$, so that

$$4\exp(-4\log s) = 4\exp\left(-\left[\frac{-\frac{(\sqrt{j-1}+1)}{\sqrt{2n}} + \sqrt{\left(\frac{(\sqrt{j-1}+1)}{\sqrt{2n}}\right)^2 + \frac{8(j-1)\epsilon}{3n}}}{\frac{4(j-1)}{3n}}\right]^2\right)$$

$$4\log s = \left[\frac{-\frac{(\sqrt{j-1}+1)}{\sqrt{2n}} + \sqrt{\left(\frac{(\sqrt{j-1}+1)}{\sqrt{2n}}\right)^2 + \frac{8(j-1)\epsilon}{3n}}}{\frac{4(j-1)}{3n}}\right]^2.$$

For the equation (4.15), $\epsilon = \alpha\frac{\log s}{n} + \beta\sqrt{\frac{\log s}{n}}$. We simplify the above equation,

$$-\frac{(\sqrt{j-1}+1)}{\sqrt{2n}} + \sqrt{\left(\frac{(\sqrt{j-1}+1)}{\sqrt{2n}}\right)^2 + \frac{8(j-1)\epsilon}{3n}} = 2\sqrt{\log s}\left(\frac{4(j-1)}{3n}\right).$$

Squaring both sides yields

$$\left(\frac{(\sqrt{j-1}+1)}{\sqrt{2n}}\right)^2+\frac{8(j-1)\epsilon}{3n}=\left(2\sqrt{\log s}\left(\frac{4(j-1)}{3n}\right)+\frac{(\sqrt{j-1}+1)}{\sqrt{2n}}\right)^2 + \frac{(\sqrt{j-1}+1)^2}{2n}.$$

Extracting the terms on the RHS, we have

$$\frac{\left(\sqrt{j-1}+1\right)^2}{2}+\frac{8(j-1)\epsilon}{3}=4\log s\frac{16(j-1)^2}{9n} + (16\sqrt{\log s})\left(\frac{(j-1)}{3}\right)\left(\frac{\sqrt{j-1}+1}{\sqrt{2n}}\right)+\frac{(\sqrt{j-1}+1)^2}{2}.$$

Substituting ϵ in the above equation yields

$$\frac{\left(\sqrt{j-1}+1\right)^2}{2}+\frac{8(j-1)\{\alpha\frac{\log s}{n}+\beta\sqrt{\frac{\log s}{n}}\}}{3}=4\log s\frac{16(j-1)^2}{9n} + (16\sqrt{\log s})\left(\frac{(j-1)}{3}\right)\left(\frac{\sqrt{j-1}+1}{\sqrt{2n}}\right)+\frac{(\sqrt{j-1}+1)^2}{2}.$$

Simplifying the above equation, we get

$$\frac{\log s}{n}\left[\frac{8(j-1)}{3}(\alpha)-\frac{64(j-1)^2}{9}\right]+\frac{\log s}{n}\left[\frac{8(j-1)(\beta)}{3}-\frac{16(j-1)}{3\sqrt{2}}[\sqrt{j-1}+1]\right] = 0. \tag{8.1}$$

The above equation (8) is only true for $\alpha=\frac{8(j-1)}{3}$ and $\beta=\sqrt{2}(\sqrt{j-1}+1)$. This helps us bound $\mathsf{E}[N_{k,j}(s)]$. ∎

8.2 Proof of Lemma 7

PROOF Let us first expand both the term $\hat{g}_j$ and g_j according to the (6.6)

$$\hat{g}_j - g_j = \frac{\hat{F}_\tau(j)}{\sum_{i=1}^{j}[1-\hat{F}_\tau(i-1)]} - \frac{F_\tau(j)}{\sum_{i=1}^{j}[1-F_\tau(i-1)]}.$$

Taking the summation inside the brackets in the denominator, the above equation becomes

$$\frac{\hat{F}_\tau(j)}{[j-\sum_{i=1}^{j}\hat{F}_\tau(i-1)]} - \frac{F_\tau(j)}{[j-\sum_{i=1}^{j}F_\tau(i-1)]}$$

By making a common denominator, we have

$$\frac{\left(j-\sum_{i=1}^{j}F_\tau(i-1)\right)\cdot\hat{F}_\tau(j) - \left(j-\sum_{i=1}^{j}\hat{F}_\tau(i-1)\right)\cdot F_\tau(j)}{\left(j-\sum_{i=1}^{j}\hat{F}_\tau(i-1)\right)\cdot\left(j-\sum_{i=1}^{j}F_\tau(i-1)\right)}$$

Let us extract the terms, by multiplying, we get

$$\frac{\left[\hat{F}_\tau(j)\cdot j - \hat{F}_\tau(j)\cdot(\sum_{i=1}^{j}F_\tau(i-1))\right] - [j\cdot F_\tau(j) - F_\tau(j)\cdot(\sum_{i=1}^{j}\hat{F}_\tau(i-1))]}{j^2 - j\sum_{i=1}^{j}\hat{F}_\tau(i-1) - j\sum_{i=1}^{j}F_\tau(i-1) + \sum_{i=1}^{j}\hat{F}_\tau(i-1)\sum_{i=1}^{j}F_\tau(i-1)}$$

$$= \frac{j\cdot\left(\hat{F}_\tau(j)-F_\tau(j)\right) - \hat{F}_\tau(j)\cdot\left(\sum_{i=1}^{j}F_\tau(i-1)\right) + F_\tau(j)\cdot\left(\sum_{i=1}^{j}\hat{F}_\tau(i-1)\right)}{j^2 - j\left(\sum_{i=1}^{j}\hat{F}_\tau(i-1) + \sum_{i=1}^{j}F_\tau(i-1)\right) + \sum_{i=1}^{j}\hat{F}_\tau(i-1)\sum_{i=1}^{j}F_\tau(i-1)}$$

Adding and Subtracting the term $F_\tau(j)\left(\sum_{i=1}^{j}F_\tau(i-1)\right)$ yields

$$\frac{j\left(\hat{F}_\tau(j)-F_\tau(j)\right) + \sum_{i=1}^{j}F_\tau(i-1)\left[-\hat{F}_\tau(j)+F_\tau(j)\right] + F_\tau(j)\left[\sum_{i=1}^{j}\hat{F}_\tau(i-1) - \sum_{i=1}^{j}F_\tau(i-1)\right]}{j^2 - j\left(\sum_{i=1}^{j}F_\tau(i-1) + \sum_{i=1}^{j}F_\tau(i-1)\right) + \sum_{i=1}^{j}\hat{F}_\tau(i-1)\sum_{i=1}^{j}F_\tau(i-1)}$$

From (6.7), we can write that with probability at least $1-\delta$, the above expression is upper bounded by

$$\frac{j\left(\epsilon_j\right)+\sum_{i=1}^{j} F_\tau(i-1)\left[\epsilon_j\right]+F_\tau(j)\left[j\cdot\epsilon_{j-1}\right]}{j^2-j\left(\sum_{i=1}^{j}\hat{F}_\tau(i-1)+\sum_{i=1}^{j}F_\tau(i-1)\right)+\sum_{i=1}^{j}\hat{F}_\tau(i-1)\sum_{i=1}^{j}F_\tau(i-1)}$$

Adding and Subtracting the term $j\left(\sum_{i=1}^{j}\hat{F}_\tau(i-1)\right)$ in the denominator yields,

$$\frac{j\epsilon_j+\sum_{i=1}^{j}F_\tau(i-1)\epsilon_j+F_\tau(j)\left[j\epsilon_{j-1}\right]}{j^2-j(\sum_{i=1}^{j}\hat{F}_\tau(i-1)+\sum_{i=1}^{j}F_\tau(i-1))+\sum_{i=1}^{j}\hat{F}_\tau(i-1)\sum_{i=1}^{j}F_\tau(i-1)+j\left(\sum_{i=1}^{j}\hat{F}_\tau(i-1)\right)-j\left(\sum_{i=1}^{j}\hat{F}_\tau(i-1)\right)}$$

$$\leq\frac{j\left(\epsilon_j\right)+\sum_{i=1}^{j}F_\tau(i-1)\left[\epsilon_j\right]+F_\tau(j)\left[j\epsilon_{j-1}\right]}{j^2-j\left(\sum_{i=1}^{j}\hat{F}_\tau(i-1)+\sum_{i=1}^{j}F_\tau(i-1)-\sum_{i=1}^{j}\hat{F}_\tau(i-1)\right)-j\sum_{i=1}^{j}\hat{F}_\tau(i-1)+\sum_{i=1}^{j}\hat{F}_\tau(i-1)\sum_{i=1}^{j}F_\tau(i-1)}$$

$$\leq\frac{j\epsilon_j+\sum_{i=1}^{j}F_\tau(i-1)\left[\epsilon_j\right]+F_\tau(j)\left[j\epsilon_{j-1}\right]}{j^2-j\left(\sum_{i=1}^{j}\hat{F}_\tau(i-1)+\left[j\epsilon_{j-1}\right]\right)-j\sum_{i=1}^{j}\hat{F}_\tau(i-1)+\sum_{i=1}^{j}\hat{F}_\tau(i-1)\sum_{i=1}^{j}F_\tau(i-1)}$$

Since the cumulative density function is a measure of probability, it is bounded between $[0,1]$. Using this, we can upper bound the above expression as

$$\frac{j\epsilon_j+\sum_{i=1}^{j}\min\{\hat{F}_\tau(j)+\epsilon_j,1\}\left[\epsilon_j\right]+\left[j\epsilon_{j-1}\right]\min\{\hat{F}_\tau(j)+\epsilon_j,1\}}{j^2-j\left(\sum_{i=1}^{j}\hat{F}_\tau(i-1)+\left[j\epsilon_{j-1}\right]\right)-j\sum_{i=1}^{j}\hat{F}_\tau(i-1)+\sum_{i=1}^{j}\hat{F}_\tau(i-1)\sum_{i=1}^{j}\max\{\hat{F}_\tau(i-1)-\epsilon_{i-1},0\}} \tag{8.2}$$

■

The above (8.2) gives us the right quantity to form a confidence radius $a_j(s)$ around $\hat{g}_j$ for $N_j(s)$ pulls of arm j

Bibliography

References

Abernethy, J. D., Amin, K., and Zhu, R. (2016). Threshold bandits, with and without censored feedback. In Lee, D., Sugiyama, M., Luxburg, U., Guyon, I., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc.

Agarwal, A., Foster, D. P., Hsu, D. J., Kakade, S. M., and Rakhlin, A. (2011). Stochastic convex optimization with bandit feedback. In *Advances in Neural Information Processing Systems*, pages 1035–1043.

Auer, P., Cesa-Bianchi, N., and Fischer, P. (2002). Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, 47(2):235–256.

Badanidiyuru, A., Kleinberg, R., and Slivkins, A. (2013). Bandits with knapsacks. In *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*, pages 207–216. IEEE.

Bernstein, S. (1934). Teoriia veroiatnostei [The theory of probabilities]. *Moskva–Leningrad: Gosudarstvennoe Tekhniko-Teoreticheskoe Izdatel'stvo.[2nd augmented ed. The 3rd ed., of the same year, is identical. The 4th ed., augmented, appeared in 1946.]*.

Bhatia, R. and Davis, C. (2000). A better bound on the variance. *The American Mathematical Monthly*, 107(4):353–357.

Blackwell, D. (1946). On an equation of Wald. *The Annals of Mathematical Statistics*, 17(1):84–87.

Cohen, A., Hazan, T., and Koren, T. (2016). Online learning with feedback graphs without the graphs. In *International Conference on Machine Learning*, pages 811–819.

Ding, W., Qin, T., Zhang, X.-D., and Liu, T.-Y. (2013). Multi-armed bandit with budget constraint and variable costs. In *Twenty-Seventh AAAI Conference on Artificial Intelligence*.

Dvoretzky, A., Kiefer, J., and Wolfowitz, J. (1956). Asymptotic Minimax Character of the Sample Distribution Function and of the Classical Multinomial Estimator. *The Annals of Mathematical Statistics*, 27(3):642 – 669.

Flajolet, A. and Jaillet, P. (2017). Logarithmic regret bounds for bandits with knapsacks. *arXiv preprint arXiv:1510.01800v4*.

Gal, S., Landsberger, M., and Nemirovski, A. (2007). Participation in auctions. *Games and Economic Behavior*, 60(1):75–103.

Grimmett, G., Grimmett, G. R., and Stirzaker, D. (2001). *Probability and random processes*. Oxford university press.

Jain, L. and Jamieson, K. (2018). Firing bandits: Optimizing crowdfunding. In *International Conference on Machine Learning*, pages 2206–2214. PMLR.

Joulani, P., Gyorgy, A., and Szepesvári, C. (2013). Online learning under delayed feedback. In *International Conference on Machine Learning*, pages 1453–1461.

Komiyama, J., Sato, I., and Nakagawa, H. (2013). Multi-armed bandit problem with lock-up periods. In *Asian Conference on Machine Learning*, pages 100–115.

Lai, T. L. and Robbins, H. (1985). Asymptotically efficient adaptive allocation rules. *Advances in Applied Mathematics*, 6(1):4–22.

Lattimore, T., György, A., and Szepesvári, C. (2014). On learning the optimal waiting time. In *International Conference on Algorithmic Learning Theory*, pages 200–214. Springer.

Li, L., Jamieson, K., DeSalvo, G., Rostamizadeh, A., and Talwalkar, A. (2017). Hyperband: A novel bandit-based approach to hyperparameter optimization. *The Journal of Machine Learning Research*, 18(1):6765–6816.

McAfee, R. P. and McMillan, J. (1987). Auctions with entry. *Economics Letters*, 23(4):343–347.

Samuelson, W. F. (1985). Competitive bidding with entry costs. *Economics letters*, 17(1-2):53–57.

Sharoff, P., Mehta, N., and Ganti, R. (2020). A farewell to arms: Sequential reward maximization on a budget with a giving up option. In Chiappa, S. and Calandra, R., editors, *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pages 3707–3716. PMLR.

Stegeman, M. (1996). Participation costs and efficient auctions. *Journal of Economic Theory*, 71(1):228–259.

Tran-Thanh, L., Chapman, A., Rogers, A., and Jennings, N. R. (2012). Knapsack based optimal policies for budget–limited multi–armed bandits. In *Twenty-Sixth AAAI Conference on Artificial Intelligence*.

Weed, J., Perchet, V., and Rigollet, P. (2016). Online learning in repeated auctions. In *Conference on Learning Theory*, pages 1562–1583.

Xia, Y., Ding, W., Zhang, X.-D., Yu, N., and Qin, T. (2016). Budgeted bandit problems with continuous random costs. In *Asian Conference on Machine Learning*, pages 317–332.